Plantel Aguascalientes
Acuerdo No. 1857 del 8 de abril del 2015 del Instituto de Educación del Estado de Aguascalientes

Tesis.
Para obtener el grado de Maestría en Ciencia de los Datos y Procesamiento de Datos Masivos (BIG-DATA)

Título de la Tesis:

Análisis predictivo en Twitter para detectar patrones de personas con tendencia Hacktivista aplicando Big Data, Machine Learning y Deep Learning.

Presenta:
Edwin Gerardo Gómez Hernández

Director:
Dr. Iván Castillo Zúñiga

México – Colombia, Enero de 2022

ASUNTO: Carta de liberación de tesis.

Aguascalientes, Ags., 31 de enero de 2022.

LIC. ROGELIO MARTÍNEZ BRIONES
UNIVERSIDAD CUAUHTÉMOC PLANTEL AGUASCALIENTES
RECTOR GENERAL

P R E S E N T E

Por medio de la presente, me permito informar a Usted que he asesorado y revisado el trabajo de tesis titulado:

"ANÁLISIS PREDICTIVO EN TWITTER PARA DETECTAR PATRONES DE PERSONAS CON TENDENCIA HACKTIVISTA APLICANDO BIG DATA, MACHINE LEARNING y DEEP LEARNING"

Elaborado por EDWIN GERARDO GÓMEZ HERNÁNDEZ, considerando que cubre los requisitos para poder ser presentado como trabajo recepcional para obtener el grado de MAESTRÍA EN CIENCIA DE LOS DATOS Y PROCESAMIENTO DE DATOS MASIVOS (BIG-DATA).

Agradeciendo de antemano la atención que se sirva a dar la presente, quedo a sus apreciables órdenes.

ATENTAMENTE

Dr. Iván Castillo Zúñiga
Nombre y firma del Director de tesis

Tabla de contenido

Índice de figuras

Índice de Tablas

Resumen

El crecimiento del uso de las tecnologías y su uso en campañas políticas pueden volver virales ciertos asuntos de gobierno y llegar a más personas, lo cual, ha vuelto muy vulnerables a representantes de la politica con personas que tienen pensamientos diferentes y que de una u otra manera quieren afectar a las personas con protestas contra una idea, lo que causa ataques a estos miembros políticos, tanto en sus redes sociales (realizando hacktivismo) cómo de manera personal.

A través de las redes sociales las personas publican información que puede ser usada para el análisis de esta y aportar soluciones a estos problemas, un claro ejemplo el Hacktivismo, el cual ha crecido cada vez más, en la búsqueda de hacer vulnerables a ciertos dirigentes políticos.

En esta investigación, se pretende desarrollar un modelo que permita la identificación de vocabulario hacktivista, a través de la combinación de técnicas de data mining y algoritmos de machine learning y Deep learning, el resultado será la implementación de un modelo con bastante precisión en la identificación y clasificación del vocabulario que refiere a Hacktivismo.

Palabras Claves: Hacktivismo, machine learning, Deep learning, data mining, corpus lingüístico.

Abstract

The growth in the use of technologies and the use of these for political campaigns and the desire to make certain political issues viral to reach more people, has made these very vulnerable with people with different thoughts and who in one way or another want to reach people with protests against an idea, which causes attacks on these political members, attacks both on their networks and personally.

Through social networks, people publish information that can be used for the analysis of this and provide solutions to these problems, a clear example of Hacktivism, which has grown more and more, in the search to make certain political leaders vulnerable

In this research, it is intended to develop a model that allows the identification of hacktivist vocabulary, through the combination of data mining techniques and machine learning and deep learning algorithms, the result will be the implementation of a model with enough precision in the identification and vocabulary classification that refers to Hacktivism.

Keywords: Hacktivism, machine learning, Deep learning, data mining, linguistic corpus.

Agradecimiento

A Dios todo poderoso, que pone en mi camino las personas necesarias para cumplir mi propósito, mis padres, mi esposa y mi hija, que son mi motor para seguir adelante, para soñar con nuevas metas y llegar a cumplirlas.

Dedicatoria

A Dios todo poderoso por permitirme compartir mi vida con mi esposa, que me acompaña y me ofrece un resguardo en los momentos de dificultad y en los momentos de alegría. Ella es quién me alienta cuando estoy cansado y me da ánimos cuando entristezco, me apoya a seguir adelante.

Introducción

De acuerdo con (Gómez & Farrera, 2019), cada vez es más frecuente que aparezcan noticias, estudios o informes sobre algún problema de seguridad en Internet y todos alcanzan niveles impacto global a la seguridad de la información de las personas e instituciones. Los ataques adquieren relevancia inmediata, activan protocolos de protección y seguridad, y llega a una opinión pública amplia.

El hacking y los delitos informáticos han sido una de las primeras amenazas que hicieron despertar la necesidad de regularizar el uso y el abuso de las herramientas tecnológicas e internet. Para finales dela década de los 90 según Comesaña, (2017), comenzó la proliferación de un conjunto de actividades relacionadas con el hacking, pero que hacían énfasis en su utilización como medio de denuncia de actividades relacionadas con la proliferación nuclear, combinando el humor y las herramientas tecnológicas con el fin de transmitir un mensaje político, social y ambiental. Un nuevo término sería acuñado para identificar este tipo de actividades: Hacktivismo

La problemática que se presenta en la investigación es la detección de características o patrones de conducta de personas con tendencia Hacktivista a partir de la información obtenida de la cuenta pública de Twitter del actual presidente de Colombia Iván Duque. Como bien se sabe, existen muchas investigaciones acerca de cómo extraer datos de Twitter con minería de datos, con el fin de tomar los datos y analizarlos para tomar la mejor decisión en diferentes aspectos. Para el caso de este estudio, se tendrá

un enfoque en buscar la mejor implementación de Big Data para clasificar las características de las personas Hacktivistas con el objetivo de determinar patrones que se concluyan como determinantes para definir este tipo de conductas.

El argumento para iniciar esta investigación, se dio a partir de la necesidad de explorar beneficios que se pueden obtener con la información de una red social, con el objetivo de detectar características de personas Hacktivistas, a través de técnicas como la minería de datos, procesamiento del lenguaje natural y la implementación de algoritmos de machine learning.

Para este estudio, se obtendrán resultados sustentados tanto de manera teórica como práctica, a través de la combinación de distintas técnicas de minería de datos, la creación de un corpus lingüístico de características de Hacktivismo y el uso de distintos modelos de aprendizaje supervisado y no supervisado (establecidos en lenguaje R y Python) para la debida clasificación de la información obtenida de Twitter.

Es importante mencionar que la organización de la tesis, esta basado en la metodología de Hernández-Sampieri, et al. (2014), considerando 6 capítulos: 1. Planteamiento del problema, 2. Marco teórico, 3. Método, 4. Resultados y pruebas, 5. Discusión y 6. Conclusiones, finalmente las referencias bibliográficas en formato APA.

Capítulo I Planteamiento del problema

En el presente capítulo se presenta como introducción, la definición del problema enfocado en el descubrimiento de patrones sobre personas que indiquen una tendencia hacia el hacktivismo en las redes sociales. Adicional a esto, se plantean los objetivos de la investigación, la justificación enfocada a generar nuevas estrategias para detectar este fenómeno en las redes sociales, la hipótesis del trabajo, que nos permitirá fundamentar los resultados tomando en cuenta la experticia de otros investigadores sobre este tipo de problemas sociales. Además, se incluye una descripción muy breve de cada uno de los capítulos en el trabajo.

1.1. Formulación del problema

1.1.1 Contextualización

El Hacktivismo hace referencia al empleo de métodos tecnológicos y especializados para presionar o conseguir objetivos relacionados con la política. La palabra Hacktivismo nace como acrónimo de la palabra Hacker y Activismo. Se ha convertido en una de las formas más efectivas para transmitir un mensaje ideológico en la red. En otras palabras, el Hacktivismo podría definirse como el uso de las herramientas tecnológicas legales o ilegales con fines políticos.

Internet es, en efecto, el medio que más dramáticamente alteró las comunicaciones personales, y sin duda alguna, alteró el ritmo de la crisis en los medios de comunicación para todos los actores políticos. En la actualidad, la comunicación por internet ocupa un papel fundamental para todo el mundo, así que darle un significado a la relación entre los gobernantes y los internautas es una prioridad muy alta para cualquier mandato. A principios de la década de los noventa, cuando los primeros actos Hacktivistas aparecieron en el mundo, era muy frecuente que desde el ámbito académico interpretaran estos fenómenos como una expresión natural del activismo político en donde los hechos eran protagonizados por redes de ciudadanos.

Para tener una mejor perspectiva acerca del nacimiento del Hacktivismo nos relata García (2018), en la década de los setenta del pasado siglo cuando los hackers de entonces deciden romper con la dinámica imperante de ocultación y privatización del

software y luchar por la liberalización del mismo, como una paso previo e imprescindible para combatir el cibercontrol social y batallar por la libertad del conocimiento y la justicia social.

Un informe realizado por el centro de Análisis y creatividad de las Tic (Tic Tac), reveló que entre marzo y diciembre de 2020 hubo un incremento del 101% en las denuncias realizadas por ciberataques y más de 37.000 reportes de noticias criminales instauradas ante la Fiscalía General de la Nación de Colombia.

A partir de esa información y de que esta problemática ha venido en aumento de una manera asombrosa, sobre todo en el último año, debido a las condiciones sociales actuales implantadas por causa del Covid 19, es necesario identificar con herramientas tecnológicas, los indicios de este tipo de activismo cibernético y posibles actos en donde los ciudadanos busquen atacar, manipular o derrocar un ente político. Sin duda alguna, la cantidad de datos a analizar es inmensa, por lo que es necesario utilizar las tecnologías de Big Data y Machine Learning para soportarla y hacerla práctica.

1.1.2 Definición del problema

Peña Castañeda (2016), relata que en entrevista con EL TIEMPO, Alberto Samuel Yohai, presidente de la Cámara Colombiana de Informática y Telecomunicaciones, dijo que para los criminales ahora es mucho más sencillo actuar de manera digital que presencial.

Con una primera observación a nuestros entornos sociales, las noticias, y multiplicidad de tareas cotidianas, podemos inferir que Internet tiene una importancia central que organiza el sistema de información en las sociedades actuales, y por tanto es clave realizar estudios de fenómenos en la red que posibiliten conocer el nuevo orden, funcionamiento y comprensión de una multiplicidad de fenómenos sociotécnicos relacionadas a la red (Gómez & Farrera, 2019).

Las redes sociales, pertenecientes a la Web 2.0, le dan un énfasis a lo social al permitir que se conecten grupos de personas en torno a gustos en común. Por tanto, se sabe que las redes sociales se han convertido en una herramienta muy utilizada para las movilizaciones sociales realizadas por distintos grupos y movimientos en los últimos años. Los hacktivistas adoptan estrategias y herramientas más directas y transgresoras que las usadas por el ciberactivismo, puesto que creen que sus tácticas de confrontación son más eficaces que las fórmulas convencionales (García, 2018).

El mismo presidente de la república hace referencia en una declaración reciente en donde menciona: debemos reconocer que hoy en día estamos en la era de la pos verdad y vemos noticias falsas por todas partes y, dada la manera en que nuestros mundos se están creando especialmente en cuanto a las redes sociales, estas pueden profundizar las divisiones en la sociedad. Algunos de los demagogos y populistas del mundo van a tomar ventaja de esto para exacerbar el discurso del odio, pero también exacerbar la violencia y muchas otras maneras de expresión que no necesariamente son positivas para las democracias.

Expuesto lo anterior, es importante mencionar, que la presente tesis se enfoca en el análisis de información sobre la red social Twitter con el propósito de detectar características o patrones con tendencias al hacktivismo. Como caso de estudio, seguidores de la cuenta del presidente de la republica de Colombia Iván Duque.

Problemática de manera específica:

1. Dificultad para transformar la información que proviene de Twitter a datos estructurados y establecer las características para asociarlo al Hacktivismo.
2. Dificultad para descubrir tendencias, patrones y correlaciones desconocidas en los datos, así como la clasificación de los mismos.
3. Dificultad en la precisión para la detección de patrones o características Hacktivistas a partir de algoritmos de aprendizaje supervisado y no supervisado.

1.1.3 Pregunta de Investigación

¿Qué dificultades existen en la construcción de un corpus lingüístico para clasificar los datos a partir de la información de las redes sociales?

¿Es posible clasificar intenciones maliciosas o activistas dentro de un contexto político, en donde se implementen modelos predictivos y descriptivos de aprendizaje supervisado y minería de datos en una red social?

1.2. Justificación

Como se viene abordando en el presente trabajo de investigación, el Hacktivismo siendo relativamente moderno en lo que respecta al uso y manejo del internet, es también un acto que ha venido tomando cada vez mayor fuerza y que al analizar sus implicaciones a nivel sociopolítico, ha dejado también grandes consecuencias o secuelas en el entorno social de cualquier nación, al utilizar el internet como una herramienta cívica y que además está siendo manejada por personas con amplios conocimientos tecnológicos. “Los nuevos movimientos sociales se van a caracterizar por una gran informalidad y espontaneidad en su formación y estructura, utilizando los medios digitales como una herramienta de difusión de mensajes y propulsora de acciones concretas” (García, 2018, pág 144). Luego de la revolución en el mundo digital se observa un margen más alto en la participación de los ciudadanos y, por ende, en el tipo de protestas que se desarrollan en las diferentes redes. Así es como poco a poco se ha ido acrecentando el número de personas que hacen movilizaciones sociales, o acciones sociales concretas y más específicas como el Hacktivismo.

Es por tal motivo que es necesario realizar procesos de identificación y análisis de los comportamientos de los cibernautas y de esta manera poder identificar con mejor precisión qué personas o grupos sociales desarrollan esta tendencia y de tal forma prevenir posibles acciones en contra de los entes políticos que puedan afectar el adecuado desarrollo social. Así, hay casos sin terminar y diferentes ejemplos de personas y acciones que han dado un vuelco a la red virtual y la han sujetado paralelamente a la

vida real, donde las personas no solo encuentran lo que deseen, conocen lo que no creían, sino que también toman acciones que en la vida real afectarían el doble de lo que podría garantizarse o evidenciarse en el mundo virtual.

1.2.1. Conveniencia

Algunos miembros del hacktivismo respaldarían sus actos refiriendo que su base para fundamentar el hacktivismo “parte de una conciencia colectiva y adquiere una actitud comprometida socialmente poniendo sus conocimientos al servicio de la ciudadanía y promoviendo políticas tales como la libertad de expresión, los derechos humanos y la ética de la información” (García, 2018).

No obstante, es importante considerar que grandes actores políticos se han visto notablemente afectados por dichos movimientos y especialmente por los actos colectivos que han puesto en riesgo los procesos electorales.

1.2.2. Relevancia social

Actualmente, tras cada uno de los sucesos que pueden establecer los ataques Hacktivistas en el ámbito político, resulta de mucha importancia el monitoreo constante de los usuarios de las redes sociales que interactúan directa o indirectamente con los perfiles de los nombres políticos, para detectar posibles ataques cibernéticos que puedan poner en riesgo los diferentes medios de comunicación digitales que se poseen. Por esto,

tras la investigación que se realiza, se busca identificar conductas que indiquen posibles ataques cibernéticos analizando el lenguaje natural con que las personas se comunican en las redes sociales, en este caso de estudio particularmente la red social Twitter.

1.2.3. Implicaciones educativas

Este trabajo investigativo fortalece las bases y las columnas teóricas para futuros estudios similares o por complemento de otras investigaciones que deseen profundizar en cómo identificar patrones de conducta en redes hacktivistas y que por medio de los hallazgos se logre pre visualizar una red de movimientos y se prevengan futuras fracturas en el ejercicio político del internet.

Así mismo, propone una visión general y contextualizada en la cual se desarrolla el hacktivismo y desarrolla una perspectiva que permite la identificación de casos en los que se deba realizar un análisis más contundente.

1.2.4. Relevancia teórica

El fenómeno del hacktivismo, como se mencionó anteriormente, es relativamente nuevo y las áreas o el campo de las ciencias sociales no lo ha abordado o investigado a profundidad, por lo cual se convierte en un enfoque necesario y relevante en el campo social, de manera que el observar y estudiar este movimiento hacktivista permitirá ir constituyendo más aportes al pensamiento.

En ese sentido, permitirá ir comprendiendo aquellos patrones distinguibles en personas con dicha tendencia y la forma en cómo dichos grupos sociales se conectan y generan movilizaciones que eluden los procesos naturales y por medio del conocimiento tecnológico y la sed de libertad sociales, desarrollen hacktivismo, alterando los comportamientos sociales fruto de la lucha política por medio del social media.

1.2.5. Utilidad metodológica

La elaboración y aplicación de modelos de machine learning en la detección de conductas con tendencias al hacktivismo, además de las técnicas de minería de datos, una vez sean demostrada su validez y confiabilidad de resultados podrán ser utilizados en otros trabajos de investigación y en instituciones educativas.

1.3. Objetivos

1.3.1 Objetivos general

Establecer un modelo predictivo que logre identificar y categorizar las características de los Twits con contenidos Hacktivistas, a través de la implementación de algoritmos desarrollados que nos aporten el conocimiento basados en una ontología semántica que clasifique dicha información.

1.3.2 Objetivos específicos.

1. Determinar el conjunto de datos a través de la extracción de Twitter, enfocados en los Retweets cuenta pública del actual presidente de Colombia Iván Duque.
2. Implementar técnicas de aprendizaje supervisado para la detección de características de personas Hacktivistas a través de lenguajes de programación (Python) basados en Machine Learning
3. Realizar el análisis de los resultados que presenta cada algoritmo en el proceso de clasificación y realizar la comparación con demás trabajos realizados y determinar la efectividad del modelo.

1.4. Viabilidad

Este trabajo reúne características, condiciones técnicas y operativas con un enfoque de desarrollo de algoritmos, los cuales, con el conocimiento obtenido en el transcurso de la maestría y la experiencia, aseguran el cumplimiento de las metas y los objetivos establecidos en cuanto a las extracciones de información con minería de datos y construcción de procesos de machine learning con clasificación de lenguaje natural.

1.5. Hipótesis

Las técnicas de Big Data en combinación con los modelos de Procesamiento del Lenguaje Natural (PLN), Machine Learning y Minería de Datos, favorecerán en la clasificación de características de personas con una conducta Hacktivista y así mismo, se podrá identificar los algoritmos más contundentes en la detección de características Hacktivistas en Twitter.

1.6. Motivaciones de la investigación

Teniendo en cuenta la problemática planteada y dado que la Maestría en Ciencia de los Datos y Procesamiento de Datos Masivos (BIG-DATA) de la Universidad de Cuauhtémoc (Campus Aguascalientes), busca formar maestrantes con los conocimientos y habilidades necesarias para analizar y procesar grandes volúmenes de datos con el propósito de aplicar los resultados obtenidos en diferentes áreas de desempeño. Por esto, se plantea esta investigación con el fin de desarrollar una solución de Big Data para detectar características o patrones que identifiquen una tendencia Hacktivista en las personas que interactúan con la cuenta pública de Twitter del actual presidente de Colombia Iván Duque, aplicando técnicas de procesamiento de lenguaje natural, algoritmos de clasificación y predicción con Machine Learning. En este mismo sentido, la motivación personal, para crear nuevas estrategias con el fin de tener herramientas de

recopilación de información, de procesamiento, de análisis y de apoyo para contrarrestar los ataques Hacktivistas.

1.7. Breve descripción de la tesis

La estructura de la tesis está formada por seis capítulos, además de las referencias bibliográficas y anexos.

El capítulo 1, Planteamiento del problema, tiene como objetivo proveer al lector del contexto de la investigación, resumiendo el problema de estudio atacado, las aportaciones previas en el tema de investigación, definición de conceptos clave y todo aquello que contribuya al entorno mencionado al inicio de este.

El capítulo 2, Estado del arte, se describen los antecedentes de la investigación, los trabajos relacionados con la problemática que aborda el proyecto actual. Se establecen las bases teóricas en las que se está sustentando el trabajo de tesis, se enfoca en conceptos relacionados con Big data, minería de datos, procesamiento del lenguaje natural, aprendizaje supervisado y aprendizaje no supervisado. Por último, se aborda de manera puntual el tema de investigación para este caso el Hacktivismo.

El capítulo 3, materiales y métodos, presenta una descripción detallada de la propuesta de solución a la problemática planteada y a los objetivos establecidos al inicio de la investigación.

El capítulo 4, pruebas y resultados, se muestran y explican los productos obtenidos tras la evaluación y validación de la propuesta de la solución desarrollada.

El capítulo 5, discusión, se presenta el análisis de los resultados de la investigación, una comparación con trabajos relacionados, la discusión respecto a los objetivos planteados y por último la respuesta a la hipótesis planteada.

El capítulo 6, conclusiones, se detallan las ventajas y las desventajas de la investigación, considerando la problemática y los objetivos establecidos al inicio del estudio.

Capítulo II Marco teórico

En el presente capítulo se aborda el Estado del arte, en donde se describen los antecedentes de la investigación, las investigaciones o trabajos relacionados con el problema que aborda la investigación actual, los cuales proporcionan un horizonte y un punto de partida para establecer las posibles soluciones a la problemática planteada. Así mismo, se establecen las bases teóricas en las que se sustenta la tesis, en donde se describen conceptos relacionados como Big Data, Minería de Datos, Procesamiento del Lenguaje Natural, Procesamiento de Datos, Machine Learning, Aprendizaje supervisado.

2.1. Marco conceptual

En el desarrollo del estado del arte se presentan los conceptos a abordar Big Data, Minería de Datos, Análisis de datos, Machine Learning, Aprendizaje Supervisado y Hacktivismo, los cuales son los temas principales del marco conceptual y que se deben tener claros para entender el desarrollo de la presente investigación

2.1.1 Big Data

El termino Big Data se refiere a los datos que son imposibles de procesar con los métodos tradicionales porque son demasiado grandes, rápidos o complejos. La naturaleza compleja del Big Data principalmente se debe a la gran parte de los datos generados no estructurados por las tecnologías modernas.

“El acto de acceder y almacenar grandes cantidades de información para la analítica ha existido desde hace mucho tiempo. Pero el concepto de Big Data cobró impulso a principios de la década de 2000 cuando el analista de la industria, Doug Laney, articuló la definición actual de grandes datos como las tres V:” (SAS (2021))

Las 3 Vs se definen de la siguiente manera:

- Volumen: El volumen se refiere a la escala de los datos, por ejemplo, cientos de TB para muchas empresas, que es parte de los que hace que los grandes volúmenes de datos sean masivos.
- Velocidad: Se refiere tanto a la velocidad con que los datos se crean, como a la velocidad con la que se procesan, se almacenan y se analizan.
- Variedad: La variedad se refiere a las diferentes fuentes de datos que pueden existir, por ejemplo, sensores, posts de redes sociales, transacciones bancarias, ventas y marketing entre otros.

2.1.2 Minería de datos

De acuerdo con Bermeo (2018), La minería de datos es un proceso dentro del campo de la estadística y las ciencias computacionales, que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Extrae información subjetiva oculta en un texto mediante algoritmos y técnicas de análisis clasificándolo como positivo o negativo. Dicho proceso puede ser aplicado en texto cortos, como los más de 500 millones de Tweets por día que escriben cerca de 100 millones de usuarios activos en Twitter.

Así mismo, vemos que la disponibilidad de muchas herramientas de minería de datos, como R, Python, RapidMiner, WEKA entre otros; abre una gama de opciones para ser aplicadas en el desarrollo de aplicaciones usadas para la segmentación de clientes o para impulsar el marketing, además del análisis de tendencias que se puede desarrollar.

2.1.3 Machine Learning

Machine learning es una forma de la inteligencia artificial que permite a un sistema aprender de los datos en lugar de aprender mediante la programación explícita. En definitiva “el Machine learning es un maestro del reconocimiento de patrones, y es capaz de convertir una muestra de datos en un programa informático capaz de extraer inferencias de nuevos conjuntos de datos para los que no ha sido entrenado previamente”. (BBVA ,2019)

La importancia del Machine Learning en la actualidad yace a partir de que la minería de datos y otros métodos de extracción de información crecen cada vez más y pueden llegar procesar grandes volúmenes de datos sin ningún análisis y la variedad de estos datos amplia que el ojo humano tardaría mucho tiempo en analizar.

2.1.4 Aprendizaje supervisado

Para el desarrollo de la investigación mediante minería de datos de Twitter, se deben aplicar diferentes métodos de aprendizaje, lo cual necesita la implementación de técnicas similares al aprendizaje humano a partir de experiencias de los distintos programas que asimilan el comportamiento de los patrones.

Los modelos predictivos reciben instrucciones sobre lo que necesitan aprender y cómo es que deben aprenderlo, a esto se le conoce como aprendizaje supervisado. “Infiere una función a partir de una serie de ejemplos etiquetados para posteriormente predecir una salida para otro conjunto distinto de ejemplos no etiquetados, de esta forma es como el algoritmo va aprendiendo a clasificar las entradas comparando con el modelo ya entrenado con sus etiquetas”. (Barrientos & Mamani, 2019). Algunos métodos y algoritmos del aprendizaje supervisado: Naive Bayes, Neural Network, Decision Tree, Logistic Regression, Support Vector Machine.

2.1.5 Análisis de datos

De acuerdo con (Treviño & Rivera & Garza ,2020), El análisis de datos se ha convertido en una de las llaves principales en la permanencia en el mercado empresarial. Además (Treviño & Rivera & Garza ,2020), continúa mencionando que las organizaciones deben entender el análisis de datos como un puente que conduce hacia la comprensión y la correcta interpretación de las necesidades que están teniendo los consumidores y que se ven reflejadas a través de datos capaces de ser interpretados.

La analítica de datos ordena y organiza la información para que estos sean utilizados en métodos que ayuden a explicar los sucesos históricos y así predecir el futuro. Para analizar los datos es necesario la disposición alta de tiempo, ya que se debe preparar los conjuntos de datos y dejarlos listos para el procesamiento de los mismo a través de herramientas que logren la obtención del conocimiento.

2.1.6 Hacktivismo

El Hacktivismo hace referencia a la utilización no violenta de herramientas digitales persiguiendo fines políticos, estás herramientas pueden ser desfiguraciones de páginas web, redirecciones, ataques de denegación de servicios, robo de información, suplantaciones virtuales, desarrollo de software, entre otros.

Profundizando en el tema, Gómez & Farrera (2019) definen "el Hacktivismo como una forma emergente de acción social que pretende cuestionar y transformar el orden social existente a través del activismo tecnológico. Dicho activismo, se sustenta en los principios de la socialización del conocimiento, la cooperación tecnológica y la autogestión comunicacional". Así mismo concluyen diciendo que "el Hacktivismo es una forma de ecología política y un nuevo movimiento social" y así lo vemos cómo sinónimo de activismo tecnológico, aportando que "no es una política que se limite a operar mediante la retórica; sus prácticas acompañan la construcción de discursos".

Las actividades Hacktivistas se extienden a diferentes tipos de ideas y causas políticas, aunque el termino sigue siendo muy controvertido, algunos mencionan que se acuño para describir cómo las acciones directas en las tecnologías de comunicación e información podían usarse a favor del cambio social al combinar la programación con el pensamiento crítico. Otros utilizan el término como sinónimo de actos maliciosos y destructivos que vulneran la seguridad del internet como plataforma tecnológica, económica y política.

2.2 Marco legal

Fueron revisadas las Leyes, Decretos, Normas o Acuerdos del ámbito Nacional o Internacional que tienen relación con los ejes temáticos de esta investigación como la seguridad de las personas, los delitos, las iniciativas para prevenir o castigar acciones delincuenciales, la protección de los datos personales y el uso de datos abiertos.

2.2.1 Ley 1273 de 2009

El congreso de Colombia (2009) modifica el código penal colombiano, se crea un nuevo bien jurídico tutelado denominado de la protección de la información y de los datos y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones.

El capítulo 1 refiere sobre los atentados contra la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas informáticos. Y el capítulo 2 sobre los atentados informáticos y otras infracciones.

2.2.2 Ley 1712 de 2014

Esta es la Ley de Transparencia y del Derecho de Acceso a la Información Pública, regula la potestad que tienen todas las personas para conocer y acceder a la información

pública generada o controlada por los sujetos obligados (presidencia de la República, vicepresidencia, ministerios, superintendencias, Consejo Superior de la Judicatura, Corte Suprema de Justicia, Corte Constitucional, Consejo de Estado, Gobernaciones, Alcaldías, etc.), a través de dos mecanismos: desde la página web de la entidad donde se debe publicar una información mínima obligatoria o mediante una solicitud (Congreso de Colombia, 2014).

Esta ley impulsa el uso de datos abiertos como mecanismo efectivo para que la sociedad civil, las entidades públicas, la academia y otros grupos de interés realicen procesos de investigación y control social a los entes gubernamentales en torno a diversos temas. En la actualidad la plataforma www.datos.gov.co cuenta con más de 16.000 conjuntos de datos disponibles y cabe mencionar que de esta fuente provienen los datos que fueron usados en este trabajo.

2.2.3 Documento CONPES 3920 de 2018

El documento CONPES 3920 de 2018 estableció la política de explotación de datos (Big Data) para Colombia, siendo el primer país en Latinoamérica y octavo en el mundo (Estados Unidos en 2012; Australia, Corea del Sur, Reino Unido, Japón en 2013; Francia en 2014 y China en 2015) en instaurar una política pública integral que busca aprovechar los datos con el fin de generar proyectos orientados a impulsar el desarrollo social y económico y la productividad. La política aborda cuatro ejes centrales (Departamento Nacional de Planeación, 2018):

- Generación de datos digitales (insumo).
- Cultura de datos (demanda).
- Capital humano para la explotación de datos (oferta).
- Marco jurídico, ético e institucional orientado a la protección de los individuos en el contexto de disponibilidad y explotación masiva de datos.

2.3. Marco referencial

2.3.1 Aprendizaje automático para la detección de humor en Twitter.

Se presenta una investigación doctoral para la experimentación con diferentes algoritmos de aprendizaje automático en las siguientes tareas: búsqueda de una solución a la detección de humor de textos en español a partir de un tuit; puntuación de cuán gracioso es dicho tuit; y, por último, el estudio en Twitter de diversos personajes famosos y comprobar qué tan graciosos son a partir de los modelos desarrollados.

Para Andújar (2020) el objetivo principal consiste en el estudio y la experimentación de diferentes algoritmos de aprendizaje automático en la tarea de detección de humor de los usuarios de la red social Twitter, utilizando para ello únicamente el contenido contextual de los tuits. Así mismo, “para obtener un modelo de aprendizaje automático se requiere de datos de entrenamiento etiquetados según la clase a la que pertenezcan. En este caso, para entrenar un modelo capaz de realizar la tarea

de detección de humor los datos de entrenamiento se corresponden con tuits etiquetados como humorísticos o como no. En este proyecto los tuits escogidos para formar el conjunto de datos de entrenamiento se han recopilado de tres fuentes diferentes". (Andújar, 2020)

Este trabajo refleja algunas de las técnicas a utilizar dentro de esta tesis, aquí se ven reflejados los resultados de los diferentes algoritmos experimentados árboles de decisión, KNN, SVM, Naive Bayes, regresión logística y perceptron multicapa encontrando que el algoritmo desarrollado de regresión logística tiende a puntuar los tuits con valores similares por lo que creemos que es debido en parte al modelo de clasificación que tiene más en cuenta ciertas características como puede ser la aparición de hashtags, exclamaciones o interrogaciones y mayúsculas y suele etiquetar como graciosos tuits con elementos similares.

2.3.2 Predicción de ataques de Cyber Bullying mediante técnicas de aprendizaje profundo apoyándose en un corpus de entrenamiento para la clasificación de texto en español.

Se presenta el desarrollo de una tesis, cuyo objetivo es preparar una base de datos de expresiones obtenidas de Twitter, con el fin de usarlo como base esencial para el entrenamiento de una red convolucional (CNN), mediante técnicas de aprendizaje profundo. El resultado de dicho entrenamiento, generará un modelo de predicción de

textos que pueden o no presentar signos de cyber acoso, en donde se manifiesten agresiones verbales, insultos, ataques racistas, homofóbicos, entre otros (Cumba, 2018).

Esta tesis desarrollada por (Cumba, 2018), se establece una arquitectura basada en fuentes de datos, que es donde se define la obtención de la información para el entrenamiento del modelo, y se tomaron datos de Twitter ya que Ecuador contaba en su momento con 800 mil usuarios activos; continuando con la arquitectura se presenta la fase de extracción y clasificación de datos, la cual define el esquema de extracción de los mismos usando un script desarrollado en Python que interactúa con Twitter a través de una API, revisando y clasificando de manera manual los datos. Una vez que se extraen y se clasifican los datos, se pasó a la preparación, en donde se creó un script en Python que se encarga de bloquear el ingreso de palabras o símbolos que ocasionaran ruido en el entrenamiento del modelo. Por último, el aprendizaje profundo, el cual define el proceso para realizar el entrenamiento de la Red Neuronal y así obtener el modelo de predicción, para ello se utilizó Python y el módulo de inteligencia artificial Keras.

Para concluir con este trabajo, podemos decir que, al desarrollar la extracción, el procesamiento y el entrenamiento del algoritmo Deep learning se generó un vocabulario de cyber bullying, el cual se etiquetó en bullying y no bullying, logrando una efectividad del algoritmo del 98%, con el que se permite garantizar el desarrollo de un sistema que permite prevenir y alertar casos de Cyber Bullying.

2.3.3 Automatic identification of suicide notes from linguistic and sentiment features.

En el presente artículo, se busca implementar algoritmos de aprendizaje supervisado, para identificar automáticamente notas de Suicidio en la Web, diferenciándolos de otros mensajes basado en un análisis de sentimientos y características lingüísticas (Schoene & Dethlefs, 2016).

Se utilizaron tres (3) conjuntos de datos para el análisis de las notas suicidas, un corpus de notas genuinas de Suicidio y dos corpus para realizar la comparación de los datos; estos últimos fueron datos recogidos del público utilizando el sitio Web de Experience Project. El corpus de notas de Suicidio genuino (GSN), fueron recolectados de diversas fuentes, incluidos artículos de periódicos y corpus existentes de otros recursos académicos. El segundo corpus es el de Amor/Felicidad (LH), la cual fue recogida en 142 publicaciones con temas relacionados al amor, alegría, entre otras, lo que se buscó fue determinar emociones, y, por último, el corpus de Depresión/Soledad (DL), el cual se recopiló dado que este puede estar cerca de las emociones y en el uso del lenguaje del corpus GSN, por lo tanto, se puede establecer diferencias de personas deprimidas y que comunican un sentimiento suicida (Schoene & Dethlefs, 2016).

Para (Schoene & Dethlefs, 2016), las características del corpus se basaron en la encapsulación de emociones, como miedo, culpa, desesperanza, tristeza, información, instrucción, perdón, felicidad entre otras, las cuales fueron agrupados en emociones positivas, negativas y neutras. Para lograr determinar estás características lingüísticas, se utilizaron herramientas como Python, NLTK LIWC 2 y, por último, para la

implementación a nivel de la experiencia, se usó el kit de herramientas WEKA, con el fin de realizar el entrenamiento de aprendizaje supervisado a partir de modelos comparados como Arboles de Decisión con J48, Arboles de Regresión LMT, Naive Bayes y Zero-R. El algoritmo que obtuvo mejores resultados para la clasificación de las notas de Suicidio, fue Arboles de Regresión LMT, dado que logró una combinación de sentimientos y características lingüísticas con una precisión del 86,61%.

2.3.4 Clasificación de sentimientos usando Modelos Probabilísticos, Deep Learning y Word Embeddings para textos cortos en español

El principal objetivo que plantea (Ari, 2019), es desarrollar un clasificador de texto con técnicas basadas en modelos gráficos probabilísticos y Deep Learning que permitan determinar la polaridad de los sentimientos en textos cortos para el idioma español, comparándolo con otras investigaciones.

Para el debido desarrollo de los objetivos de la investigación, se plantea seleccionar características principales del idioma español, experimentando con algunas basadas en Word Embeddings para la mejora de la clasificación, asimismo, realizando una comparación de desempeño de clasificadores basados en Conditional Random Fields y Deep Learning para clasificar las opiniones con los conjuntos de datos estándares, y posteriormente aplicándolos en casos de estudios de los datos seleccionados para su implementación.

En el proceso (Ari, 2019), describen dos secciones, en la primera se presentan dos conjuntos de datos, los cuales son Workshop SEPLN y TASS 2017; y en la segunda sección se presentan los conjuntos de datos para entrenamiento y prueba de casos de estudio recolectados en redes sociales como Facebook y Google Play. Estos conjuntos de datos se ejecutaron sobre algoritmos Conditional Random Fields, Convolutional Neuronal Network (CNN), Recurrent Neuronal Network (RNN) y la combinación de RNN y CNN. Las ejecuciones en estos algoritmos mostraron como resultado que la mejor clasificación, es al combinar RNN y CNN con 60% de exactitud con datos TASS, y un 94.7% con datos Google Play y 84.2% con datos de Facebook, concluyendo que un algoritmo combinado con Deep Learning, logra una mejor clasificación de sentimientos.

2.3.5 Detección y Análisis de Vocabulario de Ciberterrorismo en la Web, a través del uso de Modelos Predictivos de Machine Learning

En la investigación de (Barrón, 2019), se planteó realizar un modelo para la clasificación de vocabulario relacionada con el Ciberterrorismo, obteniendo los datos de sitios Web e implementando el posible modelo a través de técnicas predictivas de aprendizaje automático y utilizando data mining.

En la investigación, el autor determina el corpus lingüístico del Ciberterrorismo, una vez se determinó el vocabulario, se procedió a realizar la búsqueda de la información en más de 1.000 sitios Web, utilizando un Crawler de rastreo, el cual se encargó de recolectar la información que contiene cada sitio Web. Una vez obtenidos los sitios, se

efectuó una estrategia genética en paralelo (ADVI) para la recuperación del vocabulario aplicando técnicas de Web Semántica y Procesamiento de Lenguaje Natural, tokenización, Stop Word, Frecuencia de Término y Frecuencia de Término Inversa.

Después de realizar la extracción de la información, se procedió a realizar la implementación de técnicas de aprendizaje supervisado en lenguaje R como, Árboles de decisión, Naive bayes. K-ésimo vecino más cercano, Redes Neuronales, Máquinas de Soporte Vectorial, Random Forest y Ada boosting; en donde se obtuvo que la mejor técnica para realizar la clasificación del vocabulario de Ciberterrorismo es la de Máquina de Soporte Vectorial, con una precisión del 97%, la cual según (Barrón, 2019) presenta un alto grado de confiabilidad.

Esta investigación nos da una idea mucho más precisa de cómo es la generación del vocabulario, las fases de búsqueda de la información, el pre-procesamiento y la utilización de técnicas Machine Learning que son más precisas para la clasificación de vocabulario Hacktivista en R, por lo tanto, puede ser una base o ruta para direccionar la presente investigación.

2.3.6 Garcés Matilla, A. (2019). Detección automática de tweets noticiosos.

Para (Garcés, 2019), las redes sociales, como Twitter, entregan información en tiempo real a miles de usuarios en todo el mundo, y gran parte de esta información que se comparte se relacionan con noticias. Es por ello que a través de este trabajo se

pretende realizar una detección automática de tweets que se relacionen con temas de noticias.

Para continuar a la implementación del algoritmo que se utiliza para la detección de lo tweets, se realizó el pre-procesamiento con el fin de limpiar los datos y tener un procesamiento óptimo. Posterior a esto se realizó un etiquetado caracterizando dos grupos de tweets, uno noticioso (P) y otro de no etiquetados (unlabeled U). Una vez que se realizó el etiquetado se propuso a través del PU-Learning construir clases negativas y positivas, con el fin de desarrollar el clasificador final de los datos.

Como resultados de la implementación a través del modelo de aprendizaje profundo, el cual se logró entrenando sólo el texto y haciendo uso de modelos Word embedding de palabras, capas convolucionales y LSTM, su precisión fue del 98%, corroborando que las aproximaciones de aprendizaje profundo tienen resultados mejores y significativo.

Continuamos validando con diferentes trabajos relacionados que es importante la incorporación de técnicas de clasificación y predicción a través de aprendizaje de máquina, asegurando que en la realización de la actual tesis se debe contemplar el uso de este tipo de técnicas.

2.3.7 An empirical analysis of machine learning models for automated essay grading

En este estudio se realiza un análisis lingüístico sobre un ensayo, y posteriormente se busca estimar la habilidad de escritura o la calidad del texto en forma de puntaje numérico o calificación de letra (Madala, Gangal , Krishna, Goyal, & Sureka, 2018).

Para (Madala et al., 2018), los sistemas AES son útiles para las escuelas, las universidades y las comunidades de empresas de pruebas para escalar de manera eficiente y efectiva la tarea de calificar una gran cantidad de ensayos. Los resultados revelan que el uso apropiado del vocabulario, la importancia de los términos en el ensayo, la coherencia entre las oraciones y los párrafos, son una fuente confiable para predecir la puntuación del ensayo. Los análisis realizados concluyen que no todas las características son iguales y pocas características son más relevantes y están mejor correlacionadas con respecto a la clase objetivo.

Los investigadores realizaron experimentos con Knn - Vecino más cercano, Regresión logística y clasificadores basados en Máquinas de Soporte Vectorial. Los resultados en 4075 ensayos en varios temas obtuvieron una precisión del 73% a 93%.

2.3.8 Detección de depresión a través de análisis textual utilizando aprendizaje automático.

Según (Orué, 2018), se busca “la creación de un modelo predictivo, el análisis, diseño e implementación de una aplicación Web usando análisis textual y aprendizaje automático” con el objetivo social de detectar de forma temprana síntomas de depresión, la cual ha sido causa de problemas de salud y discapacidad en todo el mundo. Esta investigación enfoca su problemática a personas con depresión, con el fin de obtener características que determinen un factor de esta enfermedad.

Para esta investigación se tomaron muestras de 596 estudiantes de una universidad, a las cuales se les aplicaron herramientas llamadas “Inventario Beck II” y “Test de Frases Incompletas Sacks”, para identificar personas con depresión. (Orué, 2018), aplica TF-IDF para el procesamiento y tratamiento de datos, y la implementación de técnicas de aprendizaje automático con Máquina de Soporte Vectorial desarrollado en R. Los resultados de la implementación de este modelo tuvieron una exactitud en la detección de factores de depresión del 99.2%, siendo bastante efectivo.

Esta investigación se relaciona con la tesis actual dado que implementan una técnica Machine Learning, la cual es Máquina de Soporte Vectorial implementado bajo un algoritmo en lenguaje R, con el objetivo de clasificar características personas con

depresión, la cual es un factor que incide directamente en tomar decisiones en encontrar características Hacktivistas.

2.3.9 Diseño, compilación y anotación de un corpus para la detección de mensajes suicidas en redes sociales.

En la presente investigación de (Zafra, Gómez, & Navarro-Colorado, 2017), se desarrolla un sistema de prevención del Suicidio en la red a través de un corpus piloto de mensajes de pensamiento suicida obtenido de las redes sociales.

Para los autores, la metodología de creación del corpus lingüístico, se basó en formar 97 textos escritos en diferentes redes sociales tanto en inglés como en español, dejando como resultado un tamaño final de 7968 tokens y 225 types, incluyendo los términos stop words y de 2855 tokens, y excluyendo 1808 types. Este corpus se construyó a partir de una búsqueda y selección de textos basados en la revisión bibliográfica de foros y páginas de redes sociales tanto de Deep Web como Surface Web tratándose temas suicidas. Se obtuvo un corpus balanceado, es decir un número suficiente de muestras con tendencia suicida y otra muestra sin definir su clasificación. Una vez establecidas las muestras, se obtuvieron los mensajes más frecuentes de los usuarios más activos y se logró establecer expresiones frecuentes como “no aguanto más”, que conllevó al planteamiento de listas de palabras frecuentes relacionadas al tema del Suicidio y emociones profundas como: “Suicidio, muerte, pro suicida, perdón, olvido, depresión, insatisfacción y ayuda”; estas palabras se determinaron como semillas o

términos clave para buscar en redes sociales. Adicionalmente se agregaron palabras como hastío vital, angustia, soledad, impotencia, entre otras.

Para la detección de los mensajes en las redes sociales, los autores clasificaron la información en tipos de mensajes, que se refiere a la relación del texto con el hecho del mensaje, por ello se establecieron las categorías de auto-pro-suicida, auto-nosuicida, citas, depresión, ironía, instigador, misticidad, tristeza / melancolía e indefinido.

2.3.10 Toxicom: detección de mensajes tóxicos en medios sociales.

El presente trabajo plantea la finalidad de detectar en los medios sociales de forma automática mensajes negativos, esto a través de técnicas de Inteligencia Artificial basados en algoritmos de Machine Learning y Deep Learning (Torné, 2018).

Para la implementación del algoritmo que se entrena en este trabajo para la detección de mensajes negativos se utilizó Python, la versión 3.6.1 de Anaconda y para la visualización del código se utilizó Notebooks de Jupyter, dado que Github permite leerlos correctamente. Una vez definida la arquitectura de software para el desarrollo del algoritmo (Torné, 2018), desarrolló experimentos con los comentarios obtenidos de Wikipedia, provenientes de la competición de Kaggle, Toxic Comment Classification Challenge – Identify and classify toxic online comments, por ello, se comenzó con la limpieza de los datos, en donde se hace una visión previa de estos y almacenándolo en

una carpeta data, posteriormente, se realizó la lematización de todas y cada una de las palabras de los comentarios unificándolas para los datos de entrenamiento y pruebas.

Después de hacer la agrupación se eliminan palabras que se consideren Stop words, es decir, no tienen un valor propio para la clasificación del texto; y por último con las palabras resultantes de cada comentario se almacenarán en una lista, facilitando la lectura de los datos. Una vez completo el alistamiento de los datos y la definición del vocabulario de los comentarios, procedió a realizar entrenamientos basados en las mejores características, es decir, seleccionaron el Top N de los comentarios, aclarando que se separaron los comentarios de acuerdo si eran o no tóxicos. Para el entrenamiento se usaron algoritmos Machine Learning, como por ejemplo Regresión Logística, Naive Bayes, Máquinas de Soporte Vectorial, Árboles de Decisión y optimización de Máquinas de Soporte Vectorial con Stochastic Gradient Descent y desde los algoritmos Deep Learning, se usaron Convolutional Neural Network y Long Short-Term Memory Units. Esto con el fin de clasificar los documentos en Toxic, Severe Toxic, Obscene Threat, Insult, Identity Hate y Neutral.

Como resultado de la investigación y producto del entrenamiento y análisis de los resultados de los distintos algoritmos, el modelo más robusto es mediante ensembling, con el cual logró tener una precisión de AUC del 97.54%, esto gracias a la unión de diferentes clasificadores. Otra observación importante, es que el modelo es muy rápido, llegando a dar resultados en poco tiempo, según la cantidad de algoritmos utilizados. Adicionalmente se menciona que la desventaja del ensambling, es que este algoritmo debe ser alimentado por los demás algoritmos, es decir que, si estos se comportan de

manera adecuada en la predicción, afecta directamente el rendimiento general del híbrido.

Este trabajo, se ve relacionado directamente con la tesis actual mediante los algoritmos de machine learning y el lenguaje de programación que se utilizará, se toma este trabajo cómo base de desarrollo de la tesis actual.

2.3.11 Detección de contenido malicioso mediante técnicas de Machine Learning en las redes sociales

En este trabajo, se aborda la identificación de contenido que es considerado como malicioso y su relación con los delitos informáticos; seguidamente se revisará las bases de la construcción de un modelo de aprendizaje de Machine Learning. Para Rivero (2017) esto permitirá identificar los elementos necesarios para la recolección de datos y posteriormente mostrar los resultados del modelo y las estadísticas encontradas.

Se ha desarrollado la investigación enfocada al tratamiento de contenido con URLs, debido a que la acción de un clic por parte del usuario es directamente reconocida como una solicitud de contenido de un sitio web y en caso que éste tenga una intención maliciosa, puede hacerla efectiva. En la investigación, “se observó que la proporción del contenido aumenta notablemente durante los eventos que hacen noticia como el caso de la federación de fútbol (copa FIFA 2014) que generó más de 3 billones de publicaciones

en 32 días, donde se encontraron URLs cuyos sitios web solicitaban ingresar datos para hacer que su voto cuente en una determinada causa, lo cual podría terminar en una base de emails para enviar spam o realizar un ataque dirigido con la temática de la causa". (Rivero, 2017). La recolección consistió en obtener URLs de las publicaciones mediante expresiones regulares y consultarlas en las listas negras o blacklists, tales como: Google Safebrowsing, PhishTank, VirusTotal y WebOfTrust y Surbl.

La recolección (o web scraping) se realizó a datos públicos de páginas y grupos de Facebook. A partir de esta fuente de registros, se obtuvieron las URLs y se elaboraron estadísticas informativas.

2.3.12 Análisis Masivo de Datos en Twitter para Identificación de Opinión.

En la tesis realizada por (Olarte & Casaverde, 2020), se propuso desarrollar una herramienta de identificación de texto en Twitter basada en técnicas de aprendizaje automático y Procesamiento del Lenguaje Natural (PLN), aplicando procesamiento de texto con Análisis de Sentimientos (AS) y técnicas de Machine Learning (Máquinas de Soporte Vectorial (MSV)). Teniendo en cuenta lo anterior se desarrolló una arquitectura usando lenguaje de programación Python implementando algoritmos de MSV, Red Neuronal y Naive Bayes, a través de un modelo de aprendizaje automático que determinó una polaridad de opinión negativa o positiva en los textos de Twitter analizados.

Aquí (Olarte & Casaverde, 2020), realizan la recopilación de información en un conjunto de datos TASS, la cual se recopila en un archivo de formato XML, dado que es una colección de tweets. Posteriormente, se realiza un proceso de conversión a formato csv para manejar la información, filtrando los campos que se necesitaban para la preparación del corpus inicial.

Para la captura y colección de tweets requeridos para los experimentos, (Olarte & Casaverde, 2020), hicieron uso de Python para acceder al Data Streaming de Twitter, con la finalidad de delimitar de forma geográfica una región cuadrada para la obtención de la información y posterior análisis, esto con la ayuda de tweetpy y la librería JSON de Python que decodifica el archivo JSON ofrecido por Twitter. Para el pre-procesamiento de los datos, se implementó la supresión de palabras Stop Words, utilizando la librería NLTK (Natural Lenguaje ToolKit). Posteriormente, se realizó la normalización de mayúsculas y minúsculas, y en otros procesos adicionales la tokenización para el análisis del léxico, lematización, extracción de características, validaciones cruzadas y selección de parámetros.

En la implementación de las técnicas de clasificación, (Olarte & Casaverde, 2020) determinaron el uso de MSV, teniendo en cuenta que en las pruebas fue más efectivo que Redes Neuronales y Naive Bayes. La decisión se basó teniendo en cuenta lo que se buscaba realizar en la investigación, puesto que MSV fue desarrollada para clasificaciones binarias, la cual era perfecta para el análisis de sentimientos. Esta implementación utilizó la librería de sklearn, la cual permitió un 81% de predicción sobre

los datos propuestos en el corpus para la realización del análisis de sentimientos positivos y negativos de los tweets.

Esta investigación, es otra muy relevante para tomar de punto de partida para el desarrollo de esta tesis, ya que tiene bastantes características similares acerca del desarrollo de los algoritmos de extracción de datos de Twitter y la clasificación de las tendencias Hacktivista.

2.3.13 Detección de los niveles de estrés y ansiedad en pilotos aplicando técnicas de Machine Learning.

Para este trabajo, Navarro (2018) se busca crear una herramienta que sea capaz de predecir los estados de ansiedad y estrés en personas. Para ello, se ha comenzado realizando un análisis de cómo actúa el organismo frente a estímulos, desde que se percibe éste hasta que se produce una respuesta por los músculos efectores. Es por ello que se ha comenzado estudiando el sistema nervioso y cómo actúa de manera autónoma en determinadas situaciones.

Según Navarro (2018) “la profesión de piloto es una de las más estresantes teniendo que tomar decisiones de decisiva importancia para el pasaje en cada jornada laboral. Muchos de ellos toman decisiones después de pocas horas de sueño, el 75 por ciento de los pilotos declaran que se encuentra fatigados continuamente. Además de la

figura del piloto, existen otros puestos fundamentales para la seguridad en la aviación tal y como se explica en el modelo de Reason”

En el desarrollo de los algoritmos, los más idóneos a la hora de determinar un modelo de predicción fueron SVM y KNN. En el caso de diferenciar entre todas las clases correspondientes a la base de datos de las emociones, los valores en cuanto al acierto en la predicción son próximos al 15 por ciento mientras que, después de realizar la hipótesis simplificativa sobre los targets, los resultados mejoran notablemente, superando prácticamente todos los algoritmos el 70 por ciento en acierto en la predicción

2.3.14 Hernández, V. (2015). Identificación de la presencia de Ironía en el Texto generado por usuarios de Twitter utilizando Técnicas de Opinión Mining y Machine Learning.

En este trabajo Hernández (2015), tiene como objetivo diseñar e implementar un modelo de clasificación que permita identificar la presencia de ironía en los textos escritos en Twitter, a través de herramientas de Opinion Mining y Machine Learning.

Para el desarrollo de la investigación se hacen uso de herramientas como OpinionZoom, la cual es una plataforma de análisis de sentimientos e ironía a partir de información textual en redes sociales y el método de clasificación de aprendizaje supervisado Naive Bayes Multinomial. Hernández (2015), plantea la generación de un corpus lingüístico a partir de la composición de 11 atributos los cuales conforman

características sintácticas, semánticas y emocionales o psicológicas, con el objetivo de poder describir ironía en texto. Este corpus se generó en categorías de textos irónicos y no irónicos, los cuales fueron validados por evaluadores humanos.

A través de la implementación de la técnica de aprendizaje Naive Bayes Multinomial se realizaron las pruebas de clasificación de los textos a través de un proceso 10-Fold Cross Validation. Esta ejecución se dividió en dos pruebas, la primera basada en un set negativo de textos extraídos de cuentas de Twitter de Prensa y la segunda prueba con los textos etiquetados por los evaluadores humanos. Al realizar las pruebas los resultados con el grupo de datos número uno, arrojo un 95.97% de precisión y para el grupo número dos, una precisión del 57.98% de precisión.

Este trabajo, Hernández (2017) brinda una forma de obtener información para la generación del corpus, utilizando dos formas, haciendo uso de herramientas existentes para crearlo y la otra utilizando evaluadores humanos para la clasificación de la información.

2.3.15 Sentiment analysis on twitter using streaming API.

En la investigación de (Trupthi, Pabboju & Narasimha, 2017), se realizó la recolección automática de un corpus para el análisis de sentimientos, creando un clasificador de sentimientos positivos, negativos y neutros a partir de información recolectada de Twitter.

Para recopilar el corpus de las publicaciones de texto, se usó la API de Twitter, formando un conjunto de tres clases de sentimientos, positivos, negativos y neutros. Para lograr esta clasificación los autores plantearon la clasificación por dos tipos de emoticones, el primero los que representaban emociones alegres, felices y los emoticones que representaban tristeza, aburrimiento, rabia entre otros.

Para realizar el clasificador de sentimiento, se utilizó el algoritmo de Naive Bayes, después de haber realizado pruebas con el SVM y el CRF, dando como resultado que el clasificador de Bayes daba mejores resultados. Por lo tanto, el entrenamiento de este corpus con el clasificador, presentaba resultados positivos, logrando clasificar los textos obtenidos de Twitter en el corpus de sentimientos determinados en la investigación, es decir en los sentimientos positivos, negativos y neutros (Trupthi, Pabboju & Narasimha, 2017)

2.3.16 Características de trabajos relacionados con el Hacktivismo

En la tabla 1, se tiene el análisis de las características de los trabajos relacionados del objetivo anterior, los cuales son usados para comparar con la investigación actual.

Para esta comparación se toman los trabajos más relevantes de cara a la investigación actual

Tabla 1. Características de trabajos relacionados en la detección de intención de Hacktivismo

Trabajos relacionados	Tipo de investigación		Orientación del estudio	Técnicas para la detección de intenciones en temas afines al Hacktivismo																		
				Fuente de datos (corpus)	Conjunto de datos							apredizaje supervisado										Herramientas de mineria de datos
	Cuantitativa	Cualitativa			Fuente de datos (dataset)	TF	Stopwords	Tokenizer	Sinónimos	NLP	Análisis de sentimientos	SVC	Árboles de decisión	KNN	J48	Naive Bayes	SVM	Regresión logistica	Redes neuronales	Redes neuronales con Deep	Random Forest	
Zafra, Gómez y Navarro-Colorado (2017)	✓	✓	Vocabulario detección mensajes suicidas en redes sociales	Foros y redes sociales	Páginas web			✓		✓										✓		Gate Developer 8.2
Cumba (2018)		✓	Vocabulario de Cyber Bullying	Twitter	Twitter				✓	✓										✓		Python - Keras
Torné (2018)		✓	Detección de mensajes tóxicos enmedios sociales	Wikipedia	Kaggle, Toxic Comment Classification Challenge	✓			✓	✓		✓	✓			✓	✓	✓		✓		Desarrollo propio
Barrón (2019)	✓	✓	Vocabulario de ciberterrorismo	Páginas web	Páginas web	✓	✓	✓		✓			✓	✓		✓	✓		✓		✓	Desarrollo propio en R
Olarte y Casaverde (2020)	✓	✓	Análisis masivo de datos en Twitter para identificación de opinión	TASS	Twitter - Hashtags		✓	✓		✓	✓	✓				✓			✓			Anaconda, Python, Scikit learn, NLTK, Pandas y Numpy
Orué (2017)	✓	✓	Clasificación de Características depresivas	Resultados de Test a personas	Aplicación de Test	✓				✓	✓						✓					Lenguaje R
Garcés (2019)		✓	Claificación de texto	Twitter	Tweets					✓									✓			PU_Learning

Capítulo III Materiales y método

En este capítulo se especifican los materiales y se definen los métodos que se utilizan para el desarrollo de la investigación, se presenta una descripción de la propuesta de la solución al problema planteado y a los objetivos establecidos.

3.1. Participantes (Población o muestra del estudio).

Los participantes son los usuarios activos de Twitter en la cuenta del presidente de Colombia Iván Duque, esta cuenta tiene 2,3 millones de seguidores, como se muestra en la figura 1, los cuales en nuestro caso son nuestra población de estudio

Figura 1. Twitter del presidente de Colombia Iván Duque.

De esta cuenta de twitter, se extraen los mensajes de Retweets que tenga el presidente en sus publicaciones, para así obtener el set de datos para la

implementación de los algoritmos de machine learning y Deep learning, de los cuales se extrajeron 31,842 tuits para la construcción del conjunto de datos de las pruebas.

3.2. Escenario

El conjunto de datos (dataset) utilizado para la investigación, fue obtenido de la red social Twitter de la cuenta del presidente de Colombia Iván Duque, a través de procesos con minería de datos mediante el lenguaje de programación Python, utilizando la librería de Tweepy (lo que garantiza la confiablidad de los datos obtenidos). Los datos extraidos, fueron almacenados en un archivo con formato estándar .CSV, en donde se contienen los Tweets y retweets exactos que se relacionan con la cuenta del presidente de Colombia. Posteriormente serán analizados con técnicas de Procesamiento de Lenguaje Natural y Machine Learning.

3.3. Instrumentos de información

Cómo se mencionó anteriormente, se extrajeron los datos desde Twitter asociando la cuenta del presidente Ivan Duque mediante minería de datos con

Python. Para estas descargas realizadas se tiene la siguiente descripción de cada una de las variables que integran el dataset en la figura 2.

Figura 2. Variables de Twitter para construcción del dataset del estudio

Nombre de la variable	Tipo de dato	Descripción
created_at	fecha	Momento de la creación del post de Twitter
geo	número	Ubicación de la persona que hizo el Tweet (puede ser nulo)
id_twiter	número	Identificador único del tweet procesado
lang	texto	Lenguaje del tweet
retweet_count	número	cantidad de retweets
source	texto	Procedencia del Tweet
text	texto	Texto del Tweet

3.4. Diseño del método

En el siguiente apartado se exponen los procesos realizados para analizar las variables relacionadas con el problema de investigación; así como los procedimientos que se adoptaron para dar respuesta a las preguntas de investigación y comprobar la hipótesis.

3.4.1. Diseño

De acuerdo con Hernández-Sampieri et al., (2014) esta investigación no es experimental por que no se realiza ningún tipo de manipulación en las variables (porque ya acontecieron), ni se controlaron ni se aleatorizaron para formar grupos, no se intervino directamente sobre el fenómeno de estudio.

3.4.2 Momento de estudio

Considerando que el propósito de la investigación era analizar los datos de Twitter de un grupo específico de personas entre las variables que caracterizan la tendencia hacktivista en la red social, extrayendo datos de un periodo de tiempo de 3 meses, se realizó un estudio transversal.

“Los diseños de investigación transeccional o transversal recolectan datos en un solo momento, en un tiempo único. Su propósito es describir variables y analizar su incidencia e interrelación en un momento dado. Es como “tomar una fotografía” de algo que sucede”. (Hernández Sampieri et al., 2014)

3.4.3 Alcance del estudio.

Según Villasís-Keever & Miranda-Novales (2016) una investigación puede clasificarse considerando cinco componentes, por esto, en este trabajo se tiene:

- Número de mediciones: estudio longitudinal ya que se realizan dos o más mediciones de las variables en diferentes periodos.

- Número de grupos en el estudio: solo se incluye uno razón por la cual es un estudio de tipo descriptivo.
- Existe o no una intervención por el investigador: se considera un estudio observacional ya que el investigador recopila datos asociados al fenómeno de estudio (hurto a personas), sin realizar ningún tipo de modificación de las variables.
- Momento o tiempo en que ocurre el fenómeno a estudiar: se considera como un estudio retrospectivo, pues el evento de interés ya sucedió y se pretende encontrar los factores que se relacionan con dicha ocurrencia.
- Forma de recolectar los datos: estudio retrolectivos, ya que se usan datos de fuentes secundarias.

3.5. Herramientas de software

Python es un lenguaje de programación que cumple con lo planteado y se viene perfilando como una opción recomendada para el desarrollo de software libre (Challenger, Díaz & Becerra , 2014).

Para implementar los modelos de machine learning y deep learning se utilizó la herramienta de desarrollo Python 3.8, específicamente para la construcción de las máquinas de soporte vectorial, regresión logística, turicreate y árboles de regresión.

3.6. Consideraciones éticas

Como se mencionó anteriormente, el dataset usado en esta investigación proviene de Twitter específicamente de la cuenta pública del presidente de Colombia Iván Duque, se cumple con lo establecido en la ley 1581 de 2012 que protege los datos personales en Colombia, ya que en ningún momento se tuvo acceso a información que permitiera identificar a ningún individuo, tampoco la ubicación exacta (coordenadas geográficas) del lugar donde se encuentra la persona.

Cabe aclarar que, para esta investigación, no fue necesaria la experimentación con seres vivos, por lo tanto, no se requiere la aprobación de comités de bioética

Capítulo IV Resultados y pruebas

En este capítulo, se muestras las pruebas realizadas con cada algoritmo de machine learning y Deep learning desarrollado en el lenguaje de programación Python, se describen los resultados importantes y comparan para determinar cuál es el mejor de ellos.

4.1. Procedimiento general del ensayo

Para la puesta en marcha y la implementación de pruebas para esta investigación, se propuso el análisis del conjunto de datos a través del procesamiento de miles de registros y se describe a continuación:

- Se desarrolló un algoritmo para extraer los Tweets o Retweets asociados a la cuenta del presidente de Colombia Iván Duque, bajo la metodología de Stream, se extraen estos en tiempo real durante 3 meses obteniendo así 31.842 Tweets

- Se construyó un dataset con las palabras asociadas a ataques que puedan ser catalogados como hacktivistas en la red social de Twitter, a partir de un desarrollo propio, se escogieron 100 Tweets al azar y se realizó una interpretación de cada uno obteniendo así 80 palabras asociadas a algún tipo de ataque al presidente. Adicionalmente, el dataset contiene la variable categorica a predecir que incluye los valores “SI” y “NO”

- Estas muestras están relacionadas a la aparición de las palabras elegidas de los 100 Tweets al azar en el total de Tweets descargados. La variable predictora distribuye los resultados “SI” como las palabras que aparecen en cada Tweet y las “NO” como las palabras que no se encuentran.

- Se desarrollaron pruebas con modelos predictivos de Machine learning, cómo lo fueron Regresión logística, Deep learning, Arboles de regresión, máquinas de soporte vectorial implementados en Python; así mismo, este conjunto de datos fue segmentado en un conjunto de entrenamiento equivalente al 70% de los datos y otro conjunto de prueba representando el 30% de la población respectiva.

- Se desarrollaron pruebas con los modelos predictivos de machine learning, cómo lo fueron Regresión logística, Deep learning, Arboles de regresión, máquinas de soporte vectorial implementados en Python, probando cuál de estos tiene mejor porcentaje en la predicción del vocabulario hacktivista.

- Se realizó un análisis de resultados donde se muestra el algoritmo que mostró mejores resultados en el porcentaje de precisión y las condiciones de la detección del vocabulario hacktivista

Cabe señalar que para la validación de las pruebas realizadas y para dar la confianza a los resultados obtenidos se aplican distintas técnicas, como, por ejemplo, validación cruzada y matriz de confusión. La validación cruzada es una técnica que se utiliza para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre daos de entrenamiento y prueba (Rodriguez, 2016).

La matriz de confusión es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje supervisado. En donde cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias de la clase real, lo cual permite ver que tipo de aciertos y errores está teniendo nuestro modelo a la hora de pasar por el proceso de aprendizaje con los datos (Hernández, Ramírez, & Ferri, 2004)

4.2 Extracción de datos de Twitter y construcción del dataset

Para afrontar el problema de investigación y dar alcance a los objetivos que se han planteado, se elaboró por medio de diferentes pasos un proceso con el propósito de cumplir las actividades que relacionan la obtención de los datos, el procesamiento de ellos y el análisis de los mismos, las cuales se describen a continuación:

4.2.1 Tratamiento de datos

A continuación, se explican los diferentes pasos o tareas realizadas para el cumplimiento de las actividades:

4.2.1.1 Algoritmo de stream de twitter para la obtención de los datos

Ya que para determinar el conjunto de datos especificado en los objetivos, es necesario extraer los comentarios que se realizan en la cuenta pública en Twitter

del presidente de Colombia Ivan Duque (Ver figura 3), se desarrolló un algoritmo en el lenguaje Python, que mediante una librería del mismo lenguaje llamada *"tweepy"* se realiza minería de datos y se conecta en tiempo real mediante Streaming, a detectar en todo momento los Twits que hacen alusión al presidente Ivan Duque, en donde luego de extraer los datos son almacenados en una base de datos para luego realizar el resto de análisis y pasos.

En este algoritmo (Fig. 3-4), se le han programado algunos filtros para realizar la minería de datos específicamente asociados al presidente Ivan Duque indicando expresiones como, por ejemplo: '@IvanDuque', '#IvanDuque', 'ivan duque'.

Figura 3. Comentarios en la cuenta pública del presidente Ivan Duque

Figura 4. Algoritmo de minería de datos para extracción de Twitter

```python
import tweepy
from pymongo import MongoClient
import json

class TweetsListener(tweepy.StreamListener):

    tweets = []

    def on_connect(self):
        print("Conectado")

    def on_status(self, status):

        # Recopila los tweets en ingles

        tweet = status._json
        # print(json.dumps(tweet,indent=2))
        self.save_tweet(tweet)

    # Función que indica si ocurre algun error con la api de Twitter

    def on_error(self, status_code):
        print("Problemas de conexion, Volvemos enseguida", status_code)

    # Función que recibe un tweet y lo almacena en la base de datos de MongoDB
    def save_tweet(self, tweet):
        palabras = ['cambiosocial', 'cambio social', 'libertad expresion', 'libertadexpresion', 'sabotaje', 'guerra', 'robo', 'publicidad', 'idea', 'ataques', 'n
        insertar = False
        tweetText = tweet['text'].lower()

        for pal in palabras:
            if pal in tweetText:
                insertar = True
                break

        client = MongoClient("mongodb://VOC:VOC@chec-apd08:27017/?authSource=VOC&readPreference=primary&appname=MongoDB%20Compass&ssl=false", retryWrites=True)
        db = client["VOC"]

        if insertar:
            db.tweets_t1.insert_one(tweet)
        else:
            db.tweets_t2.insert_one(tweet)

        tweetText = tweet['text']
        # print("Tweet Insertado")


    def returnTweets(self):
        if (self.tweets) == 10:
            return self.tweets

# Se obtienen las credenciales del Api de Tweeter desde el archivo credenciales.json
with open('/var/www/html/twitter/credenciales_twitter_edwin.json', 'r') as file:
    creds = json.load(file)
# Se instancian los objeto para realizar la cnecion con la libreria Tweepy
auth = tweepy.OAuthHandler(creds['consumer_key'], creds['consumer_secret'])
auth.set_access_token(creds['access_token'], creds['access_token_secret'])

api = tweepy.API(auth, wait_on_rate_limit=True,
                 wait_on_rate_limit_notify=True)

# Se intancia la Clase del Streming para obtener los tweetd
stream = TweetsListener()
streamingApi = tweepy.Stream(auth=api.auth, listener=stream)

    # Se filtran los tweet por la ubicacion de manizales
streamingApi.filter(
    # -75.55174778,5.02078586,-75.42031895,5.10488652 -75.67902656,4.98281425,-75.32949047,5.16627818
    # track=['CHECGrupoEPM', 'chec epm', '@CHECGrupoEPM', '#CHECGrupoEPM', 'chec grupo emp', 'CHEC Grupo EPM'],
    track = ['@IvanDuque', '#IvanDuque', 'ivan duque', 'colombia', 'Colombia'],
    # locations=[-76.2878482755,4.8025144,-74.5011966642,6.0445040611], # Coordenadas Bounding box Colombia https://boundingbox.klokantech.com
    stall_warnings=False,
    is_async=True

)
```

4.2.1.2 Automatización de proceso y preprosesamiento de los datos

Luego del algoritmo desarrollado, se instaló el proyecto en una tarea programada en un equipo personal, los cuales, son almacenados en base de datos, esta tarea se desarrolló durante 3 meses, en la figura 5 se muestra los resultados obtenidos por una interacción detectada en Twitter.

Figura 5. Resultados en base de datos


```
created_at: "Wed Dec 15 10:45:11 +0000 2021"
id: 1471068817128398848
id_str: "1471068817128398848"
text: "RT @BernardoAGuerra: El deterioro de la ciudad de Medellín, de sus pro..."
source: "<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter fo..."
truncated: false
in_reply_to_status_… : null
in_reply_to_status_… : null
in_reply_to_user_id: null
in_reply_to_user_id… : null
in_reply_to_screen_… : null
user: Object
geo: null
coordinates: null
place: null
contributors: null
retweeted_status: Object
quoted_status_id: 1470911618024448009
quoted_status_id_str: "1470911618024448009"
quoted_status: Object
quoted_status_perma… : Object
is_quote_status: true
quote_count: 0
reply_count: 0
retweet_count: 0
favorite_count: 0
entities: Object
favorited: false
retweeted: false
filter_level: "low"
lang: "es"
timestamp_ms: "1639565111797"
```

En la figura 5, vemos toda la información que se puede obtener de Twitter, de allí parte el análisis de los datos para obtener solo los datos necesarios para el cumplimiento de los objetivos.

Este algoritmo se mantuvo abierto en Stream durante 3 meses, escuchando todo lo que las personas respondían en vivo, así obteniendo cómo resultado una base de datos de 31.842 retwits.

Para el preprocesamiento de los datos durante esta fase, se orienta en mejorar la calidad de los datos, a fin de evitar errores y minimizar errores en la calidad de los datos. Los siguiente son los pasos referentes que se siguieron para este proceso:

- **Integración de los datos**

  Cada post o retwet que los usuarios ingresan en la cuenta oficial del presidente de Colombia Ivan Duque, es insertado en un nuevo registro en la misma base de datos.

- **Validación y limpieza**

  Se evidencia una gran cantidad de variables que no son de importancia, para el cumplimiento de estos objetivos o bien pueden ser importantes en otra clase de investigación, por este motivo se inicia con la depuración general de los campos útiles darle forma al dataset final de datos a trabajar.

- **Transformación**

  Se elabora otro dataset con los datos relacionados en el punto anterior el cual será el dataset de elaboración de datos para clasificación.

4.2.1.3 Análisis de los datos y generación de dataset para entrenamiento de modelos de machine learning y Deep learning

Luego de obtener los datos, se tienen en cuenta solo algunos datos que se consideran relevantes para la utilización en los modelos de machine learning y Deep learning, entre ellos y el más importante el texto que un usuario a retwiteado, estos pueden consultarse en la Figura 2.

Para la generación del dataset final, que nos serviría de entrenamiento para los modelos de machine learning y Deep learning para la clasificación, se desarrolló un método propio para la generación de este, basándonos en el análisis de Tweets para establecer 80 palabras claves que significaran algún tipo de ataque al presidente.

Para esto se tomó una muestra de 300 tweets aleatorios registrados en la base de datos y con un análisis manual uno a uno, se obtuvieron las 80 palabras que de otra forma atacan al presidente. (ver tabla 2).

Luego de obtener las palabras, se desarrolló un algoritmo básico (Figura 6), en donde se realizará un conteo en toda la base de datos generada de aparición de

las palabras en cada Tweet, en este mismo se desarrolló un método de limpieza para limpiar los datos de caracteres extraños, tildes, comas y demás que puedan afectar en la consulta de las palabras en los Tweets (Ver Figura 7), de esta manera vamos a obtener un dataset con la cantidad de apariciones de cada en palabra en cada uno de los Tweets, y así queda un dataset establecido (Ver Figura 8) para el entrenamiento de los modelos de clasificación de machine learning y Deep learning.

Tabla 2. Palabras elegidas de los Tweets analizados

apodos	insultos	comparaciones	aseveraciones
porki-ria	puto	uribista	asesino
cerdo	cagado	narcouribista	mafioso
marrano	imbécil	narcoparamilitar	corrupto
porco	lacayos	narcoterrorista	ladron
porko	lagarto	subpresidente	paraco
puerco	idiota	paramilitar	mafioso
titere	rata	narcotraficante	narco
marioneta	canijo	narcoparapoliticc	criminal
bufon	hps	guerrillero	narcoparaco
mitomano	marica		torcido
cacas	porqueria		populista
capo	mequetrefe		forajido
ilegitimo	pendejo		irresponsable
lamebotas.	hipocrita		ilegal
pelele	nefasto		genocida
maniroto	inútil		homicida
mamarracho	asqueroso		secuestrador
payaso	mentiroso		paranarco
nazi	mediocre		entrometido
	incompetente		negligente
	hijodeputa		tramposo
	malparido		desgraciado
	bobo		
	miserable		
	estupido		
	cochino		
	patetico		

Figura 6. Algoritmo de conteo de palabras en Tweets

```
client = MongoClient("mongodb://VOC:VOC@chec-apd08:27017/?authSource=VOC&readPreference=primary&appname=MongoDB%20Compass&ssl=false", retryWrites=True)
db = client["VOC"]

palabras = db.wordsTweet.aggregate([ ···
wordsJson = []

for x in palabras:
    wordsJson.append(deleteSymbols(x['word']))

dataTwitter1 = db.tweets_t1.aggregate([ ···
])

dataTwitter1 = list(dataTwitter1)


countText = 1
for text in dataTwitter1:
    wordsCount = {}
    textDef = deleteSymbols(text['text'].lower())
    countWords = 0
    flasg = False
    for value in wordsJson:

        contador = textDef.count(value)
        wordsCount[value] = contador
        if contador > 0:
            flasg = True
        # wordsCount[value] = wordsCount[value] + textDef.count(value)

        countWords = countWords + 1
        print(countWords)


    if flasg:
        wordsCount['CORRECTA'] = 'SI'
    else:
        wordsCount['CORRECTA'] = 'NO'

    print('---------------------------')
    print('INSERTADO')
    print(countText)
    print('---------------------------')

    countText = countText + 1

    db.tweets_dataset.insert_one(wordsCount)
```

Figura 7. Método de limpieza de datos

```
def deleteSymbols(Phrase):
        Phrase = Phrase.replace('á', 'a')
        Phrase = Phrase.replace('é', 'e')
        Phrase = Phrase.replace('í', 'i')
        Phrase = Phrase.replace('ó', 'o')
        Phrase = Phrase.replace('ú', 'u')
        Phrase = Phrase.replace('?', '')
        Phrase = Phrase.replace('*', ' ')
        Phrase = Phrase.replace('#', ' ')
        Phrase = Phrase.replace('\n', ' ')
        Phrase = Phrase.replace('¿', '')
        Phrase = Phrase.replace('-', ' ')
        Phrase = Phrase.replace('¡', '')
        Phrase = Phrase.replace('!', '')
        Phrase = Phrase.replace('.', ' ')
        Phrase = Phrase.replace(',', ' ')
        Phrase = Phrase.replace(':', ' ')
        Phrase = Phrase.replace('$', ' ')
        Phrase = Phrase.replace(';', ' ')
        Phrase = Phrase.replace('{', ' ')
        Phrase = Phrase.replace('}', ' ')
        Phrase = Phrase.replace('[', ' ')
        Phrase = Phrase.replace(']', ' ')
        Phrase = Phrase.replace('(', ' ')
        Phrase = Phrase.replace(')', ' ')
        Phrase = Phrase.replace('<', ' ')
        Phrase = Phrase.replace('>', ' ')
        Phrase = Phrase.replace('/', ' ')
        Phrase = Phrase.replace('/', ' ')
        Phrase = Phrase.replace('   ', ' ')
        Phrase = Phrase.replace('    ', ' ')
        Phrase = Phrase.replace('     ', ' ')
        Phrase = ''.join(c if c not in map(str, range(0, 10)) else "" for c in Phrase)

        return Phrase
```

Figura 8. Dataset de entrenamiento

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	inutil	asesino	asqueroso	bobo	bufon	cacas	cagado	canijo	capo	cerdo	cochino	cojones	corrupto	criminal	desgraciado	entrometido	estupido	forajido	genocida	guerrillero	hijodeputa	hipocrita	homicida	hps
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3. Resultados de pruebas con algoritmos

4.3.1 Modelos de Machine Learning y Deep learning

Fueron diseñados y probados varios modelos de clasificación tanto de machine learning cómo de Deep Learning: Regresión logística, Deep learning turicreate, Arboles de regresión, máquinas de soporte vectorial, siguiendo algunos de los autores del marco referencial.

Los modelos de clasificación fueron probados usando dos técnicas diferentes:

- Se dividió el dataset de forma aleatoria en dos grupos: entrenamiento (70% de los registros) y validación (30%); se evaluó y se comparó el desempeño de los diferentes modelos mediante matriz de confusión a partir de los datos de validación.
- Se aplicó la técnica de validación cruzada (Cross validation) con diez registros utilizando el dataset completo

Se usó la matriz de confusión y sus métricas asociadas para evaluar la eficiencia del modelo de clasificación (Arcitura Education Inc, 2018; Gironés Roig, Casas Roma, Minguillón Alfonso, & Caihuelas Quiles, 2017; Witten et al., 2011)

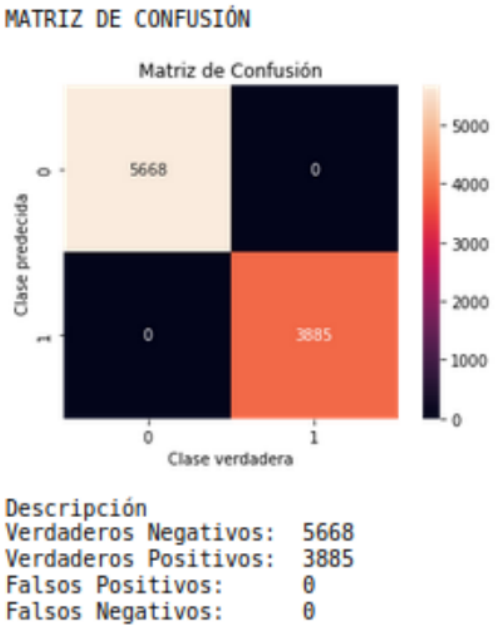
- Recall (Sensibilidad): probabilidad de obtener un verdadero positivo (número de clasificaciones correctas en la clase positiva).
- Accuracy (Precisión): proporción entre las predicciones correctas realizadas por el modelo y el total de predicciones.
- Tasa verdaderos negativos (Especificidad): probabilidad de obtener un verdadero negativo.
- Predicción positiva: proporción de registros positivos que fueron clasificados de manera correcta.
- Predicción negativa: proporción de registros negativos que fueron clasificados de forma correcta.

A continuación. se presentan los resultados de las pruebas con los modelos predictivos de regresión logística, Deep learning, Arboles de regresión, máquinas de soporte vectorial desarrollados en Python.

4.3.2 Regresión logística

En la ejecución de este algoritmo (el cual se encuentra en el anexo A.1), se aprecia en la figura 9, que el modelo predice 3885 registros, alcanzando un porcentaje de precisión promedio del 97.3% y un error del 2.7%.

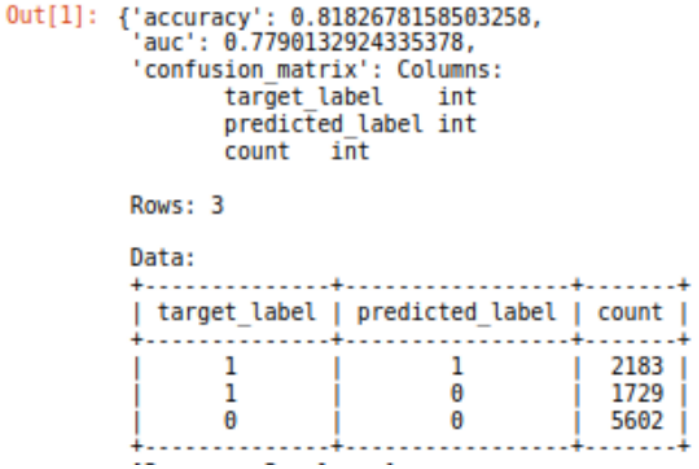
Figura 9. Matriz de confusión regresión logística Python



4.3.3 Deep learning con Python

El algoritmo de Deep learning (ver anexo A.2), cuya implementación se realizó sobre GPU, obtuvo como resultado un porcentaje de precisión promedio del 81.8%, haciendo una predicción de 9514 registros.

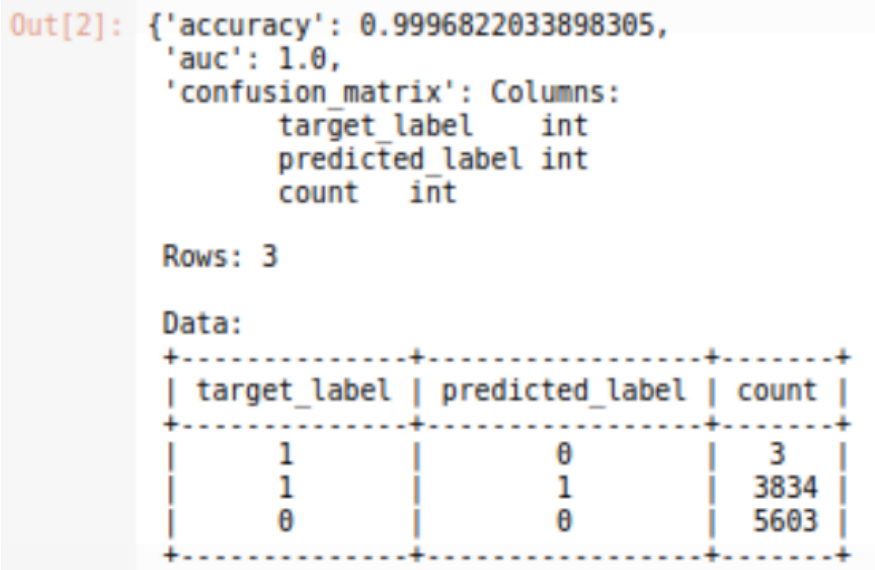
Figura 10. Matriz de confusión Deep learning utilizando GPU



4.3.4 Arboles de regresión

El algoritmo de árboles de regresión (ver anexo A.3), obtuvo como resultado un porcentaje de precisión del 99.9 % y un porcentaje promedio de error del 0.01% como se aprecia en la figura 10 y se dio en un tiempo de ejecución de 3.2 segundos.

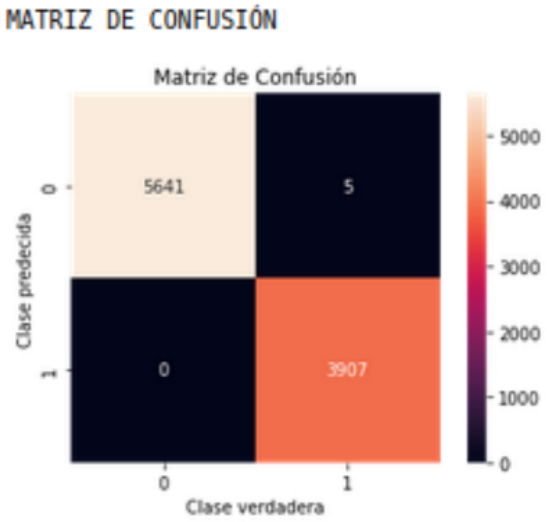
Figura 11. Matriz de confusión arboles de regresión



4.3.5 Máquinas de soporte vectorial

El algoritmo de máquinas de soporte vectorial (ver anexo A.4), obtuvo como resultado un porcentaje de precisión del 99.9% y un tiempo de ejecución de 11.1 segundos como lo podemos ver en la figura 11.

Figura 12. Matriz de confusión arboles de regresión



4.4. Comparativa de resultados

En la tabla 1, se presentan los resultados obtenidos de los algoritmos implementados en el lenguaje de programación Python con el propósito de determinar el mejor algoritmo para responder las preguntas de investigación y saber cuál es el adecuado para la predicción del vocabulario hacktivista

Tabla 3. Resultados de las técnicas de machine learning en Python

Técnica	Precisión	Error	Tiempo (seg)
Regresión logística	97.30%	2.70%	4.5
Deep learning	81.80%	18.20%	6.1
Árboles de regresión	99.90%	0.01%	3.2
MSV	99.90%	0.01%	11.1

Se puede observar en la tabla 2, que los algoritmos árboles de regresión y máquinas de soporte vectorial tiene el mismo porcentaje de precisión y de error los cuales son del 99.9% y 0.01% respectivamente y que la mayor diferencia entre ambos algoritmos de predicción son en cuanto a tiempos de ejecución, ya que las máquinas de soporte vectorial tomo 11.1 segundos y árboles de regresión 3.2 segundos con una diferencia de 7.9 segundos. Así mismo podemos ver que el algoritmo de regresión logística es muy cercano a los dos anteriores con una diferencia de 2.6 y en tiempo de ejecución de 4.5 segundos. Por último, el algoritmo de Deep learning tuvo un porcentaje de precisión bastante bajo con respeto del resto la cual fue de 81.8% con una duración de 6.1 segundos.

Capítulo V Discusión

En este capítulo, se muestra el análisis de los resultados y se realiza una comparación con las demás investigaciones que se relacionan directamente.

Se realiza una discusión respecto a los objetivos que se plantearon anteriormente y se contrastan las respuestas con las preguntas de investigación.

A continuación, se entra en discusión de los resultados que se obtuvieron en esta investigación, tomando como base los objetivos planteados y las preguntas de investigación.

5.1 Discusión del objetivo general

De acuerdo con el objetivo general y los objetivos específicos planteados en el punto 1.3, concluimos que es posible desarrollar modelos predictivos capaces de categorizar el vocabulario relacionado con el hacktivismo a partir de la cuenta oficial de Twitter del presidente de Colombia Iván Duque, apoyado en desarrollos propios de análisis de datos y algoritmos de machine learning y Deep learning.

- Se obtuvo un conjunto de datos iniciales, los cuales, eran formados por todos los Tweets y Retweets de la cuenta oficial del presidente de Colombia, con un algoritmo propio desarrollado bajo el paradigma de Stream, lo que nos permitía extraer la información en línea, de acuerdo a que las personas ingresaban los Tweets.
- Se desarrollaron 4 algoritmos de machine learning, implementados en Python, para establecer la efectividad y establecer el más indicado para la predicción del vocabulario Hacktivista.
- Se desarrollaron modelos predictivos de Deep Learning implementados bajo la tecnología GPU, estableciendo cual es la mejor.

- Se realizaron las comparaciones de los mejores modelos predictivos de Machine Learning y Deep learning, esto con el fin de que podamos determinar cuál de todas las técnicas es la mejor para la detección del vocabulario hacktivista.

5.2 Discusión con pregunta de investigación

A continuación, se da respuesta a las preguntas de investigación planteadas al inicio de este trabajo:

1. ¿Qué dificultades existen en la construcción de un corpus lingüístico para clasificar los datos a partir de la información de las redes sociales?

En un principio, esta dificultad se presentaba por la poca literatura que existe al respecto del tema del hacktivismo desde redes sociales, no existe literatura que nos pudiera identificar comportamientos, vocabulario o factores de una persona con indicios hacktivistas en las redes sociales, ni fuera de ellas.

Por estas razones, se realizó un método propio para el hallazgo de características y comportamientos de personas hacktivistas en redes sociales. Se extrajeron 300 Tweets aleatorios, los cuales se leyeron uno por uno y se identificaron grupos de palabras en forma de ataques al presidente de Colombia y

se agruparon para tener la relación del corpus lingüístico con los Tweets, se puede encontrar los grupos obtenidos en la tabla 2.

2. ¿Es posible clasificar intenciones maliciosas o activistas dentro de un contexto político, en donde se implementen modelos predictivos y descriptivos de aprendizaje supervisado y minería de datos en una red social?

En esta investigación, se demuestra que si es posible la clasificación de características de personas con tendencias Hacktivistas en redes sociales, por esto que se desarrollaron algunas técnicas de aprendizajes supervisado de machine learning y Deep learning a través del lenguaje de programación Python, donde se permitió evidenciar resultados de predicción del vocabulario superiores a 96%, demostrando así, que es posible clasificar características y tendencias de personas hacktivistas en redes sociales, particularmente en Twitter.

5.3 Discusión con trabajos relacionados

El trabajo que realizó (Cumba, 2018), tiene algunas semejanzas con esta investigación por que se implementó un modelo de deep learning y este algoritmo fue desarrollado en Python con el fin de prevenir y alertar casos de ciberbullying, al

igual que esta investigación que se desarrolló con algunos métodos de Deep learning para detectar patrones hacktivistas.

El trabajo realizado por (Ari, 2019) y esta investigación tiene algunas similitudes, que consisten en que ambos utilizaron Deep learning como técnica principal para la construcción de un modelo que permitiera clasificar y predecir el corpus creado a partir de datos obtenidos de una red social, en el caso de (Ari, 2019) obtenido de Facebook y en el caso de la presente investigación de Twitter.

El enfoque del trabajo de (Zafra, Gómez, & Navarro-Colorado, 2017) en la detección de mensajes suicidas en redes sociales, se asemeja a esta investigación ya que se obtiene un corpus con un vocabulario a partir de palabras relacionadas con Suicidio y en nuestro caso Hacktivismo, el cual en (Zafra, Gómez, & Navarro-Colorado, 2017) se planteó para ejecutarse para la búsqueda en redes sociales.

5.4 Discusión con hipótesis

Por último, conforme a la hipótesis planteada en el punto 1.5 en la que se postula que, *“Las técnicas de Big Data en combinación con los modelos de Procesamiento del Lenguaje Natural (PLN), Machine Learning y Minería de Datos, favorecerán en la clasificación de características de personas con una conducta hacktivista y así mismo, se podrá identificar los algoritmos más contundentes en la detección de características hacktivistas en Twitter.”,* de acuerdo a esto, se puede

confirmar la hipótesis planteada, donde se sustenta con la aplicación de las técnicas de aprendizaje automático, donde se combina con procedimientos de procesamiento del lenguaje natural, data mining, machine learning y deep learning, logrando generar un corpus lingüístico sobre hacktivismo, y al partir de este, se desarrollaron algoritmos de programación en Python, para tomar estos datos obtenidos y hacer pruebas determinando porcentajes de precisión positivos de la clasificación de características frases de ataques hacktivistas.

5.5 Aplicabilidad de los resultados obtenidos

Cabe resaltar, que los resultados de esta investigación pueden ser aplicables en cualquier área de estudio que se quiera abordar, no solamente para la detección de vocabulario hacktivista, sino que también puede ser utilizado para detectar otros tipos de vocabularios.

Este modelo desarrollado para la detección del vocabulario hacktivista en Twitter, puede ser utilizado por las organizaciones que trabajan en la seguridad del presidente de Colombia, con el fin de detectar ataques en la web o ataques físicos al presidente, que a través de mensajes en la cuenta pública del presidente las personas expresen la inconformidad con el presidente en forma de ataque; incluso se puede pensar, que realizando algunos ajustes básicos a la lectura de los Tweets

en tiempo real se pueda realizar detección de este tipo de vocabulario en toda la red social.

5.6 Análisis FODA de la investigación

En el desarrollo de esta investigación y debida a las diferentes actividades abordadas las cuales conllevaron al cumplimiento de los objetivos, se identificó algunas **debilidades** en el inicio que podrían afectar el desarrollo de esta, dentro de las cuales se considera la dificultad de no contar con un vocabulario asociado a tendencia hacktivista, teniendo en cuenta este como el insumo principal de los algoritmos de machine learning y Deep learning. Es por esta **debilidad** que se considera luego como una **fortaleza**, ya que al realizar la búsqueda propia de este dataset sobre 300 tweets aleatorios se genera el corpus lingüístico y la ontología semántica relacionada con el tema de investigación.

Una vez que contamos con este dataset, se centra en el desarrollo de las técnicas de machine learning y Deep learning para realizar las debidas pruebas desarrollando diferentes algoritmos bajo el lenguaje de programación Python, en donde hallamos una **oportunidad** al poder implementar los diferentes modelos que permitieron ejecutar este dataset para determinar la clasificación de las características de tendencias hacktivistas.

Una de las **amenazas** directas se encontró en la implementación de los algoritmos de Deep learning, ya que la construcción de algunos de estos, y para realizar las respectivas pruebas, se debía encontrar una arquitectura de hardware pertinente para lograr ejecutar los algoritmos satisfactoriamente, por ejemplo, en caso de algoritmos basados en Turicreate debían ser ejecutados en sistemas operativos Linux, ya que para Windows no estaba soportada la librería de Python.

Capítulo VI. Conclusión

Para este último capítulo, se describen las conclusiones sobre esta investigación, algunas ventajas, considerando la problemática y los objetivos establecidos.

La investigación, propone el despliegue de una modelo de datos adecuado para la búsqueda de información en Twitter específicamente sobre personas influyentes en la política, combinando técnicas de minería de datos, limpieza de datos, NLP (Procesamiento del lenguaje natural), eliminación de stopwords y con esto la generación de un corpus lingüístico y generación de un dataset para las pruebas de aprendizaje de máquina.

Luego de esto, la selección de los algoritmos de machine learning, como: regresión logística, Deep learning, Arboles de regresión, máquinas de soporte vectorial, de acuerdo a los mejores algoritmos asociados a otras investigaciones similares, para realizar las pruebas de detección de vocabulario hacktivista, desarrollados sobre el lenguaje de programación Python.

Con los resultados obtenidos en las pruebas realizadas, concluimos que el mejor modelo para la detección de vocabulario hacktivista es árboles de regresión ya que tuvieron un porcentaje de precisión del 99.9%, y aunque se asemeja al algoritmo de máquinas de soporte vectorial en cuanto a la precisión, se concluye que el mejor algoritmo es árboles de decisión porque el tiempo de ejecución es mucho mejor que las máquinas de soporte vectorial.

Una de las ventajas de este trabajo sobre la detección de vocabulario hacktivista, es la ventaja de haber construido un corpus lingüístico y definido una

ontología semántica bajo un desarrollo propio, teniendo en cuenta que los trabajos que se relacionaron no presentaron un vocabulario definido o se construyó a partir de libros o de sinónimos.

Otra ventaja que presenta este trabajo y que se precisa dentro de los resultados, es que se confirma en el modelo elegido de machine learning, alcanzan una precisión superior al 96% lo cual es considerado óptimo en los modelos de clasificación.

Como trabajo futuro, se propone realizar un análisis de datos distribuido, con procesamiento paralelo, estableciendo una línea para interpretar y extraer y seguir robusteciendo el corpus lingüístico desarrollado con Tweets reales y otra línea de investigación con algoritmos de machine learning y Deep learning ajustados o probando con nuevas tecnologías, esto con el propósito de tener mejor precisión en la predicción, mejor rendimiento en resultados y tiempo de ejecución de los algoritmos.

Las nuevas investigaciones, no solamente tendrían que estar basadas en Hacktivismo en redes sociales, si no que se pueden basar en otras problemáticas sociales como el acoso, violencia intrafamiliar, narcotráfico, etc. Ya que la metodología sobre la cual se desarrolló esta investigación sirve para todo lo que esté relacionado con las redes sociales.

Referencias

Alvarez, M., & Gimenez, C. (2020). “Que es python. Online]. Disponible en: https://desarrolloweb. com/articulos/1325. php.

Andújar Carracedo, Á. (2020). Aprendizaje automático para la detección de humor en Twitter (Doctoral dissertation).

Ari Mamani, D. F. (2019). Clasificación de sentimientos usando Modelos Probabilísticos, Deep Learning y Word Embeddings para textos cortos en español.

Arcitura Education Inc. (2018). Fundamental Big Data. Arcitura.

Bastidas Chacón, J. (2018). Análisis de usos y prácticas de comunicación de un grupo de activismo vegano en la red social Facebook (Bachelor's thesis, Universidad Autónoma de Occidente).

Barrientos Mogollon, E. S., & Mamani Mamani, S. A. (2019). Modelos de aprendizaje supervisado como apoyo a la toma de decisiones en las organizaciones basados en datos de redes sociales: Una revisión sistemática de la literatura.

Barrón, I. (2019). Detección y Análisis de Vocabulario de Ciberterrorismo en la Web, a través del uso de Modelos Predictivos de Machine Learning. México: Universidad de Cuauhtémoc.

BBVA (2019). 'Machine learning': ¿qué es y cómo funciona?. Obetinido de https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/

Candón-Mena, J. (2005). [Presentación e índice de] Move. Net. Actas del I Congreso Internacional Move. Net sobre Movimientos Sociales y TIC. In Move. Net. Actas del I Congreso Internacional Move. Net sobre Movimientos Sociales y

TIC (2005), p 4-8. Grupo Interdisciplinario de Estudios en Comunicación, Política y Cambio Social de la Universidad de Sevilla (COMPOLÍTICAS).

Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. Ciencias Holguín, 20(2), 1-13.

Congreso de Colombia. Ley 62 de 1993., (1993).

Congreso de Colombia. Ley 136 de 1994., (1994).

Congreso de Colombia. Ley 599 de 2000., (2000).

Congreso de Colombia. Ley 1273 de 2009., (2009).

Congreso de Colombia. Ley 1581 de 2012., (2012).

Congreso de Colombia. Ley 1712 de 2014., (2014).

Cremades, S. Z., Soriano, J. M. G., & Navarro-Colorado, B. (2017). Diseño, compilación y anotación de un corpus para la detección de mensajes suicidas en redes sociales. Procesamiento del Lenguaje Natural, 59, 65-72.

Cumba Armijos, P. D. (2018). Predicción de ataques de cyber bullying mediante técnicas de aprendizaje profundo apoyándose en un corpus de entrenamiento para la clasificación de texto en español.

FONSECA, lleana María Schmidt, et al. PELIGROS DE LAS REDES SOCIALES: EDUCANDO EN CIBERSEGURIDAD (CODIGO UNA–SIA 0585-17). En [GKA EDUTECH 2020] Congreso Internacional de Tecnologías en la Educación. 2019.

Garcés Matilla, A. (2019). Detección automática de tweets noticiosos.

García-Estévez, N. (2018). Origen, evolución y estado actual del activismo digital y su compromiso social. Ciberactivismo, hacktivismo y slacktivismo. In II Congreso Internacional Move. net sobre Movimientos Sociales y TIC (2018),

p 139-156. Grupo Interdisciplinario de Estudios en Comunicación, Política y Cambio Social de la Universidad de Sevilla (COMPOLÍTICAS).

Gironés Roig, J., Casas Roma, J., Minguillón Alfonso, J., & Caihuelas Quiles, R. (2017). Minería de datos. Modelos y algoritmos. Barcelona, España: Editorial UOC.

Gobierno de Colombia. (2019a). Plan Nacional de Desarrollo 2018-2022: Pacto por Colombia, pacto por la equidad. Recuperado de https://colaboracion.dnp.gov.co/CDT/Prensa/PND-Pacto-por-Colombia-pacto-por-la-equidad-2018-2022.pdf

Gobierno de Colombia. (2019b). Política Marco de Convivencia y Seguridad Ciudadana. Recuperado de: https://id.presidencia.gov.co/Documents/191220-Politica-Marco-Convivencia-Seguridad-Ciudadana.pdf

Gómez, D.M., & Farrera, R.A. (2019). El hacktivismo e Internet como territorio en disputa. Una mirada desde los marcos de acción colectiva. Estudios Políticos.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2014). Metodología de la investigación (Sexta). México: McGraw-Hill.

Hernández, V. (2015). Identificación de la presencia de Ironía en el Texto generado por usuarios de Twitter utilizando Técnicas de Opinión Mining y Machine Learning. Santiago de Chile: Universidad de Chile. Obtenido de http://repositorio.uchile.cl/bitstream/handle/2250/134793/Identificacion-de-lapresencia-de-ironia-en-el-texto.pdf?sequence=1&isAllowed=y

Madala, D. S. V., Gangal, A., Krishna, S., Goyal, A., & Sureka, A. (2018). An empirical analysis of machine learning models for automated essay grading (No. e3518v1). PeerJ Preprints.

Martínez Gómez, K. P. (2017). Periodismo digital y Hacktivismo: El caso de Anonymus en México. Análisis comparativo de la cobertura de la Operación Cartel en cuatro medios mexicanos.

Navarrete Bermeo, J. A. (2018). Desarrollo de una aplicación de minería de datos que permita detectar preferencias y patrones en textos y tópicos obtenidos de Twitter (Bachelor's thesis, Espol).

Navarro Cantos, C. (2018). Detección de los niveles de estrés y ansiedad en pilotos aplicando técnicas de Machine Learning.

Orué Medina, A. M. (2018). Detección de depresión a través de análisis textual utilizando aprendizaje automático, 2017.

Olarte, A., & Casaverde, A. (2020). Análisis Masivo de Datos en Twitter para Identificación de Opinión. Universidad Nacional de San Antonio Abad del Cusco, Facultad de Ingeniería Eléctrica Electrónica informática y Mecánica. Cusco, Perú: Unsaac. Obtenido de http://repositorio.unsaac.edu.pe/handle/UNSAAC/5252

Peña Castañeda, C. (2 de octubre de 2016). Los principales retos que afronta el país en seguridad informática. El Tiempo https://www.eltiempo.com/tecnosfera/novedades-tecnologia/desafios-que-afronta-colombia-en-seguridad-informatica-50410

Rivero, E. (2017). Detección de contenido malicioso mediante técnicas de Machine Learning en las redes sociales (Doctoral dissertation, Universidad de Buenos Aires. Facultad de Ciencias Económicas.).

Rodríguez, E. (s.f.). RPUBS. Recuperado el Marzo de 2020, de https://rpubs.com/eropa1981/22869

Rodriguez, O. (2016). Validación cruzada (cross-validation) y Remuestro (Bootstrapping). Recuperado el 18 de junio de 2020, de http://www.oldemarrodriguez.com/yahoo_site_admin/assets/docs/Presentación_-_CV.293124233.pdf

SAS (2021) BIG DATA Qué es y por qué es importante?. Obtenido de: https://www.sas.com/es_co/insights/big-data/what-is-big-data.html

Schoene, A. M., & Dethlefs, N. (2016, August). Automatic identification of suicide notes from linguistic and sentiment features. In Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (pp. 128-133).

Torné Alonso, R. (2018). Toxicom: Detección de mensajes tóxicos en medios sociales.

Treviño-Reyes, R., Rivera-Rodríguez, F., & Garza-Alonso, J. (2020). La analítica de datos como ventaja competitiva en las organizaciones. VINCULATEGICA EFAN, 6(2), 1063-1074.

Trupthi, M., Pabboju, S., & Narasimha, G. (2017, January). Sentiment analysis on twitter using streaming API. In 2017 IEEE 7th International Advance Computing Conference (IACC) (pp. 915-919). IEEE.

Villasís-Keever, M. Á., & Miranda-Novales, M. G. (2016). El protocolo de investigación II: los diseños de estudio para investigación clínica. Revista Alergia México, 63(1), 80–90.

Witten, I., Frank, E., & Hall, M. A. (2011). Data Mining. Practical Machine Learning Tools and Techniques (Third Edit). Massachusetts: Morgan Kaufmann Publishers.

Zafra, S., Gómez, J., & Navarro-Colorado, B. (2017). Diseño, compilación y anotación de un corpus para la detección de mensajes suicidas en redes sociales. Procesamiento del Lenguaje Natural (59), 65-72. Recuperado el 2019, de http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/5494/3253

Anexo A. Algoritmos en Python

Anexo A.1 Regresión logística

```
import turicreate as tc

# Indica el uso de GPUs en el computador
# 1 indica que usará 1 GPU, -1 utilizará todos los GPU por default y 0 indica que usara el CPU
tc.config.set_num_gpus(-1)

# Cargamos los datos recogidos en la web
data = tc.SFrame.read_csv('datasetHacktivismo.csv', sep= ';')

# Columna Correcta, con valor (Si) significa que la página Web pertenece a Suicidio
data['CORRECTA'] = data['CORRECTA'] == 'SI'

# Dividiendo el conjunto de entrenamiento y de pruebas
# 70% entrenamiento, 30% para pruebas
train_data, test_data = data.random_split(0.7)

# Construyendo el modelo de datos con el algoritmo Regresión Lógistica
model = tc.logistic_classifier.create(train_data, target='CORRECTA',
                                      features = ['inutil', 'asesino', 'asqueroso', 'bobo', 'bufon', 'cacas', 'cagado',
       'canijo', 'capo', 'cerdo', 'cochino', 'cojones', 'corrupto',
       'criminal', 'desgraciado', 'entrometido', 'estupido', 'forajido',
       'genocida', 'guerrillero', 'hijodeputa', 'hipocrita', 'homicida',
       'hps', 'idiota', 'ilegal', 'ilegitimo', 'imbecil', 'incompetente',
       'irresponsable', 'lacayos', 'ladron', 'lagarto', 'lamebotas',
       'mafioso', 'malparido', 'mamarracho', 'maniroto', 'marica', 'marioneta',
       'marrano', 'mediocre', 'mentiroso', 'mequetrefe', 'miserable',
       'mitomano', 'narco', 'narcoparaco', 'narcoparamilitar',
       'narcoparapolitico', 'narcoterrorista', 'narcotraficante',
       'narcouribista', 'nazi', 'nefasto', 'negligente', 'paraco',
       'paramilitar', 'paranarco', 'patetico', 'payaso', 'pelele', 'pendejo',
       'populista', 'porco', 'porki ria', 'porko', 'porqueria', 'puerco',
       'puto', 'rata', 'secuestrador', 'subpresidente', 'titere', 'torcido',
       'tramposo', 'uribista'])

# Guardando las predicciones del modelo de datos
predictions = model.predict(test_data)

# Evalua los resultados y guarda el modelo en un diccionario de datos
results = model.evaluate(test_data)

# Muestra los resultados finales del modelo
# Incluye porcentaje de clasificación y matriz de confusión
results
```

Anexo A. 2 Deep learning

```
import turicreate as tc

# Indica el uso de GPUs en el computador
# 1 indica que usará 1 GPU, -1 utilizará todos los GPU por default y 0 indica que usara el CPU
tc.config.set_num_gpus(-1)

import warnings
warnings.filterwarnings("ignore")

# Cargamos los datos recogidos en la web
data = tc.SFrame.read_csv('datasetHacktivismo.csv', sep = ';')

# Columna Correcta, con valor (Si) significa que la página Web pertenece a Suicidio
data['CORRECTA'] = data['CORRECTA'] == 'SI'

# Dividiendo el conjunto de entrenamiento y de pruebas
# 70% entrenamiento, 30% para pruebas
train_data, test_data = data.random_split(0.7)

# Elige automáticamente el modelo correcto en función de los datos
model = tc.classifier.create(train_data, target='CORRECTA',
                             features = ['inutil', 'asesino', 'asqueroso', 'bobo', 'bufon', 'cacas', 'cagado',
       'canijo', 'capo', 'cerdo', 'cochino', 'cojones', 'corrupto',
       'criminal', 'desgraciado', 'entrometido', 'estupido', 'forajido',
       'genocida', 'guerrillero', 'hijodeputa', 'hipocrita', 'homicida',
       'hps', 'idiota', 'ilegal', 'ilegitimo', 'imbecil', 'incompetente',
       'irresponsable', 'lacayos', 'ladron', 'lagarto', 'lamebotas',
       'mafioso', 'malparido', 'mamarracho', 'maniroto', 'marica', 'marioneta',
       'narrano', 'mediocre', 'mentiroso', 'mequetrefe', 'miserable',
       'mitomano', 'narco', 'narcoparaco', 'narcoparamilitar',
       'narcoparapolitico', 'narcoterrorista', 'narcotraficante',
       'narcouribista', 'nazi', 'nefasto', 'negligente', 'paraco',
       'paramilitar', 'paranarco', 'patetico', 'payaso', 'pelele', 'pendejo',
       'populista', 'porco', 'porki ria', 'porko', 'porqueria', 'puerco',
       'puto', 'rata', 'secuestrador', 'subpresidente', 'titere', 'torcido',
       'tramposo', 'uribista'])

# Genera y guarda las predicciones del modelo de datos
predictions = model.classify(test_data)

# Evalua los resultados y guarda el modelo en un diccionario de datos
results = model.evaluate(test_data)
```

Anexo A. 3 Árboles de regresión

```
import turicreate as tc

# Indica el uso de GPUs en el computador
# 1 indica que usará 1 GPU, -1 utilizará todos los GPU por default y 0 indica que usara el CPU
tc.config.set_num_gpus(-1)

# Cargamos los datos recogidos en la web
data = tc.SFrame.read_csv('datasetHacktivismo.csv', sep = ';')

# Columna Correcta, con valor (Si) significa que la página Web pertenece a Suicidio
data['CORRECTA'] = data['CORRECTA'] == 'SI'

# Dividiendo el conjunto de entrenamiento y de pruebas
# 70% entrenamiento, 30% para pruebas
train_data, test_data = data.random_split(0.7)
print('TURICREATE')
# Construyendo el modelo de datos con el algoritmo Arboles de Decisión
model = tc.decision_tree_classifier.create(train_data, target='CORRECTA',
                                           max_depth = 3)

# Guardando las predicciones del modelo de datos
predictions = model.predict(test_data)

# Evalua los resultados y guarda el modelo en un diccionario de datos
results = model.evaluate(test_data)

# Muestra los resultados finales del modelo
# Incluye porcentaje de clasificación y matriz de confusión
results
```

Anexo A. 4 Máquinas de soporte vectorial

```
# Librerias
# *********************************************************

# Libreria para el tiempo
import time

# Libreria para algoritmos de machine learning
import sklearn
# Libreria para trabajar con Máquinas de Soporte Vectorial
from sklearn.svm import SVC
# Importamos el train_test_split para dividir los datos de entrenamiento y prueba
from sklearn.model_selection import train_test_split
# Libreria para construir la matriz de confusión
from sklearn.metrics import confusion_matrix

# Libreria para ignorar los warnings
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)

# Libreria para cargar el dataset o dataframe
import pandas as pd
import seaborn as sns

# Libreria para usar graficación de la matríz de confusión
import matplotlib.pyplot as plt

# Especificación del algoritmo a utilizar
print("\n")
print ("-------------------------------------------------------------------------------------")
print ("                MODELO DE CLASIFICACIÓN CON MÁQUINAS DE SOPOERTE VECTORIAL")
print ("-------------------------------------------------------------------------------------")
print("\n")
print ("                        DETECCIÓN DE VOCABULARIO HACKTIVISTA E TWITTER")

# Proceso del algoritmo de Red Neuronal
# *********************************************************

# Inicia del contador de tiempo
inicio_tiempo=time.time()
```

```
# Carga el dataset
dataset = 'datasetHacktivismo.csv'
df = pd.read_csv(dataset, sep = ';')

print("\n")
print ("Dimensión del Dataset: ")
print ("Registros: ", df.shape[0])
print ("Columnas:  ", df.shape[1])
print("\n")
print ("Variables predictoras (Valores)")
arreglox = df[df.columns[:-1]].values # Especifica tomar variables predictoras
print (arreglox)
print("\n")
print ("Variable a predecir (Valores)")
arregloy= df[df.columns[-1]].values # Especifica tomar variable a predecir (última posición)
print (arregloy)

#Muestra las caracteristicas utilizadas en el análisis
print("\n")
print("Características utilizadas en el análisis")
print (df.keys())

# Especificación del modelo de datos para entrenamiento
X_train, X_test, y_train, y_test = train_test_split(arreglox , arregloy,test_size = 0.3)
print("\n")
print ("Entrenamiento: ",len(X_train), "registros (70%)")
print ("Pruebas:       ",len(X_test), "registros (30%)")

# Variable model carga el modelo de la Máquina de Soporte Vectorial
model = SVC()

# Establece en el modelo número de registros para entrenamiento y para pruebas
model.fit(X_train, y_train)

# Verifica el porcentaje de aprendizaje del algoritmo
print ("Aprendizaje del algoritmo en el entrenamiento: {0:.2f}"
       .format(model.score(X_test,y_test)),"%")
print("\n")

# Matriz de confusión de entrenamiento
y_pred = model.fit(X_train, y_train).predict(X_test)

# Matriz de confusión de pruebas
cnf_matrix = confusion_matrix(y_test, y_pred)
```

```
#Finaliza el tiempo del proceso
tiempo=(time.time()-inicio_tiempo)

# Matriz de confusión de pruebas
print("\n")
print ("MATRIZ DE CONFUSIÓN")
sns.heatmap(cnf_matrix.T, square=True, annot=True, fmt='d', cbar=True)
plt.xlabel('Clase verdadera')
plt.ylabel('Clase predecida')
plt.title('Matriz de Confusión')
plt.show()
print ("Descripción")
print ("Verdaderos Negativos: ",cnf_matrix[0,0])
print ("Verdaderos Positivos: ",cnf_matrix[1,1])
print ("Falsos Positivos:     ",cnf_matrix[1,0])
print ("Falsos Negativos:     ",cnf_matrix[0,1])

# Optimización (Parametro de C, le indica cuanto desea evitar clasificar mal un dato)
# Cuando el valor de C es más pequeño, elige un hiperplano de separación de mayor margen
# haciendo un mejor trabajo de clasificación
model_C = SVC(C=1)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)

# Cuando el valor de C es más grande, elige un hiperplano de separación de menor margen
# generando un número mayor de errores de clasificación
model_C = SVC(C=10)
model_C.fit(X_train, y_train)
model_C.score(X_test, y_test)

# Visualizando los resultados
print("\n\n")
print ("RESULTADOS DE CLASIFICACIÓN")
print ("--------------------------------------------------------------------------------")
porcentaje=(1-(y_test != y_pred).sum()/y_test.shape[0])
print("Precisión:                          {0:.1f}"
      .format(porcentaje * 100),"%")
print("Precisión optimizada (C=10):  {0:.1f}"
      .format(model_C.score(X_test, y_test) * 100),"%   **Optimiza la clasificación generando ")
print ("                                        un hiperplano de separación de menor margen")
print("Errores de clasificación:     {1} errores, sobre un total de {0} casos"
      .format(y_test.shape[0],(y_test != y_pred).sum()))
print("Tiempo de ejecución es de:    {0:.2f}"
      .format(tiempo),"seg")
```